# INFORMATION CENTRIC NETWORKING IN VEHICULAR AD HOC NETWORKS

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

**Eirini Kalogeiton**

von Athens, Greece

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik

# Information Centric Networking in Vehicular Ad Hoc Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

**Eirini Kalogeiton**

von Athens, Greece

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Der Dekan:

Bern, October, 2020

Prof. Dr. Zoltan Balogh

*Dedicated to Panagiotis…*

$\Sigma\tau o\nu$ $\Pi\alpha\nu\alpha\gamma\iota\omega\tau\eta\ldots$

# Acknowledgments

First and foremost, I want to thank my supervisor, Prof. Torsten Braun, for allowing me to pursue a Doctoral degree under his supervision. Professor's Braun guidance and support taught me to think before acting and allowed me to grow both personally and professionally. He allowed me to explore my own ideas by providing a supportive and independent research environment. I would like also to express my gratitude to Prof. Vassilis Tsaoussidis. He proposed, encouraged and supported my decision to pursue a PhD, as well as provided me with valuable comments and suggestions for this thesis. Moreover, I would like to thank Prof. David Bommes for agreeing to be the chair of my defence in such a short notice.

I want also to thank all my colleagues of the CDS group and from the INF: Dr José Carrera, with whom we shared an office over the last years, Dr Eryk Schiller who loves to learn and his feedback was always very helpful, Mostafa Karimzadeh with whom we laughed so much during the last years, Ali Marandi who taught me and made me feel helpful, Mikael Gasparyan with whom we have very interesting conversations, Jakob Schärer who always supported me and Dr Zhongliang Zhao who helped me during my first year! And to all others: Negar Emami, Alisson Medeiros, Diego Oliveira Rodrigues, Jonnahtan Saltarin, Hugo Santos, Marcel Stolz, Allan Mariano de Souza, Dimitrios Xenakis; thank you for the lessons, the support during difficult times and the of course for all the laughs we had! A big thank you to the institute's system administrator Dr. Peppo Brambilla, which gave me major help in several critical cases.

During my PhD, I was fortunate to collaborate with two other groups. Our collaboration, discussion and engagement to excellent standards was something new to me, but now I cannot think my life without these qualities. Dr Maria Rita Palattella is a mentor; Dr Gianluca Rizzo is a force that always searching for the right questions; Dr Gomes Duarte João do Monte offered me so much help during the first years; Dr Ridha Soua is always supportive and passionate for new trends; Dr Ion Tarcanu is helpful and willing to explain; Prof. Thomas Engel, whom without I would not have had worked in this

research project. Of course, I want to thank my two colleagues: Gaetano Manzo and Antonio Di Maio: Thank you for the support and the laughs all these years. Talking to you, exchanging ideas, developing and analysing taught me so much!
Special thanks go to my mental support over the last years: Daniela Schroth. Daniela without you I would be lost. Your support, your knowledge and your eagerness to help others are qualities every one must have!
To all my friends, Maria Tsoga, Nikolas Spiliarotis, Antonis Chaidaris, Rafaela Lakmeta Gkanetsiou, Iro Theodoropoulou, Matina Makri, Stathis Katharopoulos, Panagiotis Chaloulos, (and everyone else I may forget) listening to me whining and crying over the last years: thank you for everything!
Finally, I am grateful to my family, Dora and Stelios, and my sister, Vicky, for the unconditional emotional support that provided me over the last years.
Special thanks go to Panagiotis. You lifted me up when I was down. I would not have made it without you.

*Bern, October 2020* *Ειρηνη*

# Abstract

Vehicular Ad Hoc Networks (VANETs) are characterized by intermittent connectivity, leading to disruption in their communication. The current Internet Protocol Suite supports end-to-end communication, where nodes requesting content need to know the exact address of the node holding it. Thus, to support intermittent connectivity, new architectures have to be designed and tested. Information Centric Networking (ICN) is an approach aiming at evolving the Internet architecture from host-centric to the content-centric. An implementation of ICN is Named Data Networking (NDN). NDN's main principle is that a content object can be distributed among network nodes solely on its name. This thesis proposes efficient solutions to improve the performance of NDN applications in VANETs that address the current communication challenges caused by vehicular mobility and wireless standards.

First, we study how we can reduce the number of broadcast messages in a VANET, since broadcasting of messages leads to waste of network resources (decrease of bandwidth and throughput). In our first contribution, to deal with broadcasting every message from every node, we investigate how creating unicast paths between vehicles improves the communication and the content retrieval process. By using unique identifiers on vehicles, we create routing entries targeting destination vehicles, i.e. which vehicles should receive each message. Furthermore, we install on vehicles multiple omnidirectional antennas to enable simultaneous reception and transmission of a message. This allows us to satisfy vehicular requests compared to the standard broadcast scheme. But, since omnidirectional antennas are installed on the vehicles, a message still occupies the wireless medium in all directions. Hence, in our next contribution we install directional antennas on vehicles, to further limit the dissemination area of messages, and to not occupy the channel of other vehicles outside of the spreading area of messages.

In this thesis we also study whether using deployed infrastructure that supports intermittent connectivity and resource management assists the content retrieval

process. To perform so, we use street sensors (Road Side Unit (RSU)) that act as gateways that connect vehicles in VANETs. We create two routing protocols. In the first RSUs receive and send all messages from nodes, and in the second RSUs act as a back up mechanism for nodes. Indeed, we show that with their permanent use collisions occur, leading to continuous rejection of messages. To deal with this, in our final contribution, we propose the use of Software Defined Networking (SDN). SDN offers centralized control by decoupling the network control from its forwarding functions. We use SDN to construct vehicular paths, to install rules to the forwarding tables of vehicles and to adjust the RSUs transmission power to enable their connection with the maximum number of cars, without, however, rejecting all messages.
We evaluate our algorithms using simulation tools and realistic vehicular mobility traces and we show that the solutions proposed in this thesis are efficient and assist the content retrieval of an NDN application.

**Keywords:** Vehicular Ad Hoc Networks, Information Centric Networking, Named Data Networking, Software Defined Networking

# Contents

# List of Figures

# 1 Introduction

As the number of interconnected devices grows together with the number of Data produced and shared over the Internet, the limitations that exist in its current architecture pose as a significant challenge for ensuring connectivity. Hence, a new Internet architecture has been proposed, named Information Centric Networking (ICN), which ensures Data distribution over interconnected devices.

Moreover, with the introduction of smart capabilities on devices, i.e. the support of Internet connectivity over different networks, high demand for assuring a proper Internet connection among them has been emerged. In this thesis, we study a particular mobile network, the vehicular network. Vehicular networks include vehicles, public transportation, and some infrastructure deployed on city streets. These networks have particular characteristics, such as high mobility and interrupted connections leading to unstable Internet connections among vehicles. We apply the ICN paradigm to these networks and examine its impact on the distribution of Data among them.

## 1.1 Overview

The origins of the current Internet date back about 50 years, with the U.S. military's funding of a research packet switching network named Arpanet, in 1969. Arpanet connected five sites: UCLA, Stanford, UC Santa Barbara, the University of Utah and BBN [103]. Many similar packet switching networks were created at that time from 1965 to 1980, namely: the NPL network, the Merit network and the Cyclades network. Cyclades was the first network to make the hosts, rather than the network itself, responsible for reliable delivery of Data, using unreliable datagrams and associated end-to-end protocol mechanisms [169]. In 1973, after the above projects were successfully delivering Data networking between different hosts, the Internet Protocol (IP) and the Transmission Control Protocol (TCP) were created and they are still the basis for today's Internet protocol suite.

TCP is a protocol that provides a communication scheme between the application layer, i.e. where an Internet application runs, and IP. TCP's principle is based on establishing a connection between two hosts before exchanging any actual Data. TCP is part of the transport protocol of the Internet protocol suite. One of the most important functions of TCP is the establishment of the connection through the three-way handshake function. Before exchanging any information, both end-to-end hosts have to establish a secure connection by exchanging SYN and ACK messages.

Furthermore, IP is responsible for routing and addressing each message that is transmitted through the network. In particular, IP defines headers, including IP addresses in all network packets. These addresses include the source and destination information of a packet, so it can be forwarded through the network. Until today IP protocols, have been proposed, mainly the IPv4 and IPv6 protocols, but one, in particular, is used in today's networks: IPv4.

When the Internet was created, the communication was only between static computers without addressing today's developing mobile devices. In the very late 1980s, commercial Internet service providers (ISPs) began to emerge [2]. Then, in 1992, the first mass-produced GSM phone from Nokia, named Nokia 1011 was introduced into the market. Although the smartphone concept was introduced in the 1990s, it was not until the 2000s and specifically in 2007 that the smartphones as a concept became popular with introducing the iPhone by Apple [3]. Smartphones were the first mobile devices available to the public that used the Internet. With this introduction, the demand for more robust schemes for communication using the Internet increased. Fig. 1.1 shows the number of Internet users over the last 35 years (data taken from [104]). As seen in Fig. 1.1 the number of connected users has grown over 1000% over the last 20 years, with over 55% of the world population using the Interest.

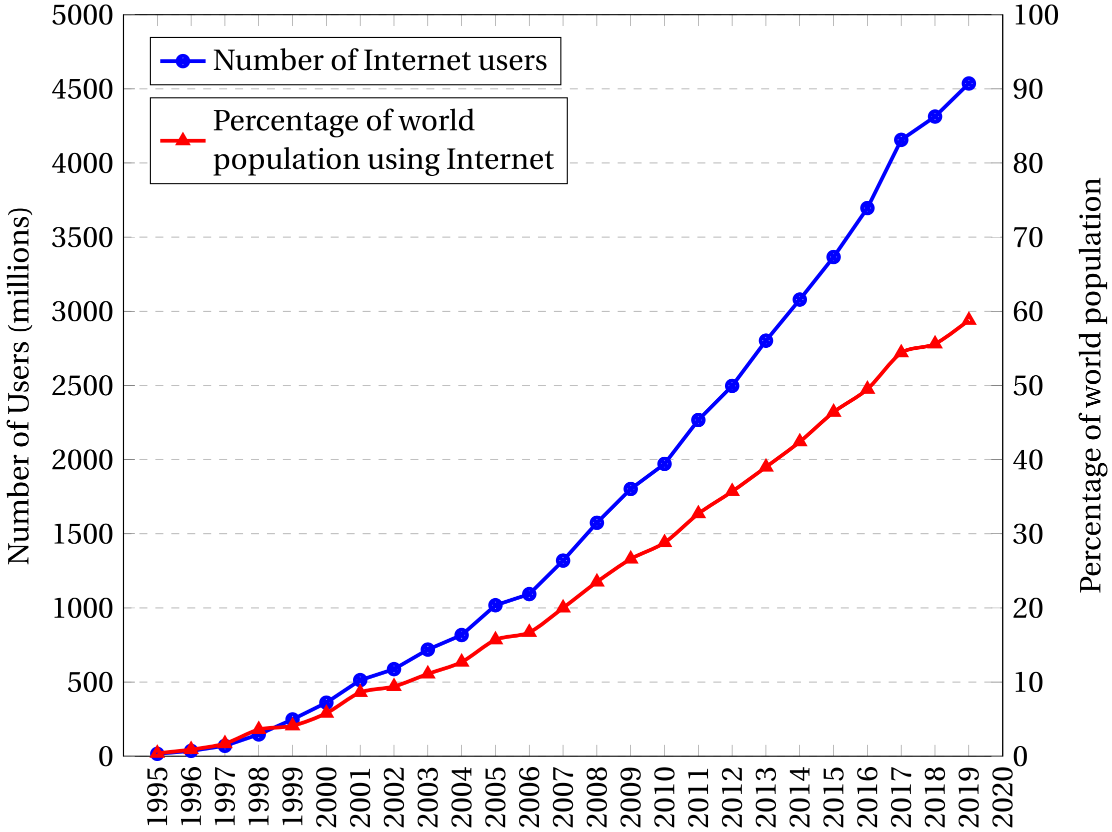



Figure 1.1: Internet Users and percentage of world population using the Internet over the last 25 years

TCP/IP is based on host end-to-end connectivity, something that challenged the communication of mobile devices, such as smartphones. The main issue TCP/IP is facing is the increasing number and the heterogeneity of interconnected devices [143]. IPv4 is not suitable to deal with this problem. Hence, IPv6 has been proposed but not used yet, since it requires the replacement or configuration of any communication device (routers, switches, etc.). Thus, over the last ten years, new network paradigms have been proposed, one of the most popular being Information Centric Networking (ICN) [45, 62, 141]. ICN addresses the limitations that IP creates by addressing the content by its name and not by the location of hosts. Hence, ICN is a significant candidate for a new era of Internet architectures, since messages are addressed via unique names and nodes can exchange their information based on these names rather than their location.

In addition, mobile users are limited by the range of transmitters or their proximity to wireless access points. To solve the above limitations, networks on-the-fly or else mobile Ad Hoc networks (MANETs) are created. Mobile Ad Hoc networks are connected usually through a wireless signal and are mostly infrastructure-less [90] by simultaneously allowing out of range nodes to route Data messages through intermediate nodes [105]. A sub-category of MANETs are Vehicular Ad Hoc Networks (VANETs). VANETs include vehicles that are equipped with on-board units (OBUs) (that are equipped with wireless devices) and sensors that are deployed on streets, such as Road Side Units (RSUs), cameras, traffic lights, etc. VANET creation is based upon the need for communication between vehicles in everyday life. VANETs use the V2X model for communication. V denotes the vehicle and X the component the vehicle is connected to. For instance, V2V denotes Vehicle to Vehicle communication, V2C denotes Vehicle to Cloud communication, V2I denotes Vehicle to Infrastructure communication, etc.

VANETs are considered being an important part of Intelligent Transportation Systems (ITS) [157]. ITS technology began in the 1970s and has been introduced to cope with the growing need for traffic management due to the steadily increasing number of vehicles. Namely, every year the number of total vehicles in Europe increases by 2% with a total of 6.2% increase only in the last 5 years [34]. It is predicted that a quarter billion of vehicles are connected in 2020 to form new in-vehicle services and automated driving capabilities [1]. VANETs provide many ITS services to vehicles [38] and are considered part of ITS since a vehicle acts as a sender, receiver, and forwarder of information [159].

Nowadays applications that are designed for vehicles can be divided mostly into two categories: infotainment and safety applications. Infotainment applications include video streaming, navigation, advertisements, while safety applications may include traffic information, road accidents, weather conditions and general traffic warnings. Moreover, VANET applications are usually related to a particular area, for instance, a navigation application of a city. Hence, information dissemination in VANETs should be performed in areas where the application content is related. Furthermore, VANETs are characterized by high mobility of nodes (node mobility usually follows a pattern [164]). Nonetheless, this mobility (even if it is predictable) disrupts the end-to-end paths that TCP/IP is based on. Hence, V2V and V2I communication is intermittent and produces high overhead, high delays and is characterised by low bandwidth, due to broadcast message transmissions. To deal with these problems, over the last years ICN has been proposed and applied in VANETs. ICN targets to change the current TCP/IP host centric communication model to a future content-centric communication scheme. In ICN, Data exchange is not based on where a host is located, but rather on which information is being requested. When applying ICN in VANETs, therefore, Data exchange is not affected by node mobility and when needed, Data can be restricted to a particular geographical area.

Moreover, in case that information is not only related to a particular geographical area, Delay Tolerant Networking (DTN) combined with ICN has been proposed [9, 124, 125]. DTN aims to deliver data in highly-dynamic wireless topologies with intermittent connectivity. DTN relies on the store-and-forward networking paradigm, allowing nodes to act as agents by caching and transferring Data among different geographical locations [125]. This allows communication between nodes and Data dissemination to different areas of the network, but requires that the information requested will have a large lifetime, i.e. it will be valid for a long duration. ICN and DTN integration in VANETs applies for several infotainment applications, e.g. when a vehicle requests the map of an area before entering this area. However, this combination would not apply in safety related applications or in infotainment applications that have a short lifetime, e.g. when requesting a real time traffic video of a road.

For implementing the ICN scheme, Content-Centric Networking (CCN) or Named Data Networking (NDN) have been proposed. They are analysed in Chapter 2.2. NDN networks are based on a reactive message retrieval process, where nodes request content to retrieve it. One main advantage of the NDN networks is also the in node caching capability; each node receiving a content object can store it in its cache.

This thesis addresses some key challenges that exist in VANETs and propose solutions for content retrieval in VANETs in a particular geographical area using the ICN and specifically the NDN paradigm. We describe these challenges together with the contributions of this thesis in the following Sections.

## 1.2 Research Questions

In this Section, we describe the research questions addressed in this thesis.

### 1.2.1 Message Broadcasting in VANETs

One of the most important characteristic of VANETs is the intermittent connectivity between nodes. This interrupted connectivity derives from the mobility of vehicles. In particular, when a vehicle moves, its connection with other vehicles in its vicinity can break. Hence, messages that vehicles send to other nodes around their vicinity may not be delivered. To address this issue of message path breaks and intermittent connectivity in VANETs, always broadcasting messages is considered the most reliable way of communication. Wi-Fi standards, such as the Wireless Access in Vehicular Environments (WAVE) standard [11], are developed and applied specifically in vehicular networks. WAVE relies on always broadcasting messages in the network. But these broadcast transmissions lead to several problems in vehicular networks, especially when the number of interconnected cars is high (e.g. in a city centre, during rush hours, etc.).

The first problem of broadcast is the waste of network resources. When many devices always broadcast messages, the bandwidth and throughput of the system decrease [60]. This is an important limitation, since the WAVE standard, which is based on IEEE 802.11p, uses channels of only 10 MHz bandwidth in the 5.9 GHz band and provides up to 27 Mbps data transmission rate in perfect conditions. Secondly, broadcasting leads to message collisions in the communication channel. This produces errors in the transmitted messages (e.g. coding errors) leading to unsuccessful message transmissions. So, nodes can neither act nor rebroadcast messages according to their routing strategy. This leads either to an increase in the number of messages in the channel or to unsatisfied content retrieval attempts (e.g. a vehicle cannot download a city map). Therefore, the first research question (**RQ1**) that is important to address is **how to reduce the number of broadcast transmissions of messages in VANETs when the number of interconnected cars is high.**

### 1.2.2 Spreading Area of Messages in VANETs

When transmitting a message through Wi-Fi using omnidirectional antennas, electromagnetic waves travel towards all directions. Even when the message transmission is unicast, the message occupies the channel according to the radiation pattern of the antenna that is installed in the device. In vehicular networks, when the number of connected cars is high because vehicles travel, it is important to target directly the destination node of the message, to avoid any unnecessary occupation of the channel, as well as to limit the nodes receiving the message. To achieve this, nodes can have directional antennas installed, or use beamforming techniques to steer their antennas towards a particular direction. Therefore, the second research question (**RQ2**) that is important to address is **how to limit the dissemination area of transmitted messages in VANETs when the number of interconnected cars is high.**

### 1.2.3 Routing Protocols Using Infrastructure for VANETs

Network architectures need to support vehicular connectivity. Although V2V is an ideal solution, in practice when the VANET is either sparse or dense, deployed network infrastructure can assist in message routing. Even though several network architectures that use infrastructure have been proposed, many of them do not consider integrating ICN. In ICN messages are exchanged according to their content and not on the location of hosts. This advantage makes ICN a crucial component to integrate into a VANET since all content applications must be able to timely react to the local demand as well as to the current physical characteristics of the vehicular network. Therefore, the third research question (**RQ3**) that is essential to study is **whether deployed infrastructure, e.g. Road Side Units, combined with ICN and appropriate routing protocols assist content retrieval in VANETs.**

### 1.2.4 Centralized V2I Communication for VANETs

With a highly dynamic environment such as vehicular networks, a decentralized communication as supported by V2V communication may not be appropriate since it creates a large amount of overhead within the network [119]. Centralized architectures in VANETs can improve the overall vehicular connectivity, as well as offer mobility and resource management mechanisms [109], by having a global network view and a unified configuration interface [57]. Moreover, a centralized architecture should be able to support efficient content retrieval policies, especially when the same content object is requested by many nodes. This can be supported by ICN, where the content is requested based on its name. Therefore, the fourth research question (**RQ4**) that is essential to study is **whether one centralized architecture combined with the integration of ICN could improve network performance, in terms of vehicular connectivity and content retrieval, in high density VANETs.**

## 1.3 Thesis Contributions

A general communication concept of VANETs is when vehicles communicate with each other (Vehicle to Vehicle- V2V communication) or with the infrastructure (Vehicle to Infrastructure - V2I communication) to deploy services (e.g. Internet access, music download) [44]. In this thesis, we use both V2V and V2I communication. Our main goal is to address the research questions RQ1- RQ4, as described in Sections Sections 1.2.1–1.2.4 to improve network connectivity and application performance in an NDN-VANET.

### 1.3.1 V2V Communication

The first part of this thesis is dedicated solely on V2V communication addressing **RQ1** and **RQ2**. We believe that a vehicle should be autonomous and not rely on any infrastructure assistance to satisfy its application requirements. In the literature, recent studies focus on communication in VANETs by presenting architectures that require large-scale networks, not only in terms of the number of vehicles but also in terms of coverage area and infrastructure [54, 57]. They focus on cloud services to establish connections between areas and to perform computations inside proposed clouds. These studies, however, do not consider small areas that vehicles can communicate with each other to perform their tasks. This Vehicle to Vehicle (V2V) communication enables offloading traffic from clouds and cellular infrastructure. We base the first two contributions of this thesis only on V2V communication.

#### 1.3.1.1 Enhanced Routing Protocols for Reducing Message Broadcasting Using V2V Communication

In our first contribution, we develop routing algorithms to **cope with the message broadcasts in VANETs** [78,84], to address the **RQ1**, as described in Section 1.2.1 . When using Wi-Fi in VANETs, every message transmission is broadcast to all connected network nodes. This provides excessive redundancy within the network by creating high overhead, limiting bandwidth and network throughput. To solve the broadcast problem, we propose a new routing protocol that avoids broadcasting requests and instead uses unicast transmissions when possible.

In particular, we assume that a content object is related only to a particular area. Then, we implement reactive routing, where the content object is available only when it is requested and we create routing entries in nodes. The entries are created from broadcasting a request until the request reaches the content source. When the content object returns to the requester, all intermediate nodes receiving the content object insert an entry into their routing tables. Hence, paths are created by identifying the next nodes that a request should be transmitted. Then, all other requests are unicast to nodes according to the routing table. Our routing protocol, therefore, allows broadcasting of requests *only* when routing entries are not available. Otherwise, messages are unicast to reduce both overhead and resource utilization in the network. In addition, we develop and analyse three main route selection techniques, and evaluate which one is the best for our network.

#### 1.3.1.2 Enhanced Solution for Limiting the Spreading Area of Messages using V2V Communication

In our next contribution [81, 83], we develop routing algorithms to **reduce the dissemination area of a message**, thus addressing **RQ2**, as described in Section 1.2.2. In particular, we start by following the idea of unicasting messages, when possible, to avoid redundant transmissions and overhead. Note that we use only V2V communication. Moreover, to further reduce the dissemination area of a message we install in each node directional antennas. Each antenna radiates its highest power towards a particular direction, allowing us to point directly to the area that we want to transmit a message. In addition, we transfer the responsibility of retransmitting a message from request-centric, i.e. only the requester retransmits a message when its Time To Live (TTL) expires, to a distributed approach, where each node is responsible for retransmitting a message when its TTL expires [81].

#### 1.3.1.3 Summary of Contributions of V2V Communication

To summarise, in the first part we address **RQ1** and **RQ2** to deal with the issue of broadcasting messages into the network and to limit the dissemination area of a message. We unicast messages, when there is an entry to the routing table, to reduce broadcasting messages. We also install in nodes directional antennas that point to particular locations and select only one directional antenna for unicast transmissions. We evaluate our protocols using network simulators. We used vehicular traces from the city centre of Luxembourg [47, 48], and compared our proposed protocols with other established routing protocols. Our results show that by limiting message broadcasts, we retrieve more content objects quicker, thus highlighting the efficiency of our network. Secondly, we show that by using directional antennas to send messages towards particular locations, we reduce the utilization of network resources.

### 1.3.2 V2I Communication

In the second part of the thesis, we provide solutions for **RQ3** and **RQ4** by using only V2I communication. We assume that a vehicle can partly rely on infrastructure that is already deployed on streets in cities. Infrastructure helps to perform necessary tasks, such as providing traffic information in vehicles, establishing forwarding rules and increasing the content availability in the VANET. As infrastructure, first, we use only Road Side Units (RSUs) that are deployed on streets. Secondly, we use RSUs and Software Defined Networking (SDN), to centralize the network, combined with NDN in a VANET.

#### 1.3.2.1 Proposed Routing Protocols Using V2I Communication

In our next contribution, we propose **routing protocols using V2I for content retrieval** [80] to study whether V2I communication can assist in message routing and content retrieval, thus addressing **RQ3**, as described in Section 1.2.3.

Our main goal is to use deployed network infrastructure and use it as part of the routing process of a message. In particular, we use infrastructure as a main or a back-up network component that is responsible for the routing of packets. In this work we use:

- already proposed data structures by NDN.
- RSUs deployed along roads.

First, we discover content sources, which then advertise their content objects back to the RSUs. We highlight that network nodes have no prior knowledge about which content object other nodes hold [80]. Our approach highlights that an easily deployed infrastructure, such as RSUs, without proper configuration, i.e. without changing its characteristics (e.g. transmission power), fails to function as the main network component and fails to manage intense traffic in case of high vehicle density. We show that due to collisions around the installed RSU, the communication fails and the RSU falls back into rejecting all messages. Nevertheless, we also highlight that using an RSU only when necessary, we manage to reduce the broadcast storm around it and effectively use it as the main network node.

#### 1.3.2.2 Enhanced Solution to Centralize VANETs using V2I Communication

In our next contribution, we address **RQ4**, as described in Section 1.2.4. Our goals are to study whether a **centralized architecture with ICN can improve network performance in VANETs**, in terms of vehicular connectivity and content retrieval. For centralizing the VANET, we use Software Defined Networking (SDN) [94, 168]. SDN is considered as one of the promising solutions that can handle the dynamic nature and dense deployment scenarios of future VANET applications [167]. Decoupling of the network forwarding functions (data plane) from the network control (control plane) together with its convenient deployment and its reliability brings potentials to offer flexibility, programmability, and centralized control knowledge. This facilitates flexible network management and control for large scale VANETs [79].

We study what impact a configurable infrastructure can have on a VANET [82]. We treat all RSUs deployed on roads as switches and we apply SDN, allowing configuration of characteristics of the RSUs via an SDN controller. In addition, we use different node configurations to examine the most beneficial configuration for the vehicles. The SDN controller is responsible for adjusting the transmission power of the RSUs that are deployed on the streets to increase the number of cars connected to them. In addition, the SDN controller performs path calculation for a request and populates the routing tables of vehicles. This allows vehicles to always have a fallback mechanism for content retrieval, without overloading the network neither with excessive broadcasts nor with downloading content from the Internet (via cellular interfaces that are costly). Therefore, except for the 1-hop broadcast of beacons in the control channel for identifying the connected cars to the RSUs, we eliminate all broadcast transmissions that are needed for content exchange and use only unicast transmissions via a particular communication channel.

#### 1.3.2.3 Summary of Contributions of V2I Communication

To summarise, in the second part we address **RQ3** and **RQ4**. We study whether infrastructure can assist in message routing and what impact a centralized architecture has on VANETs. We evaluate our protocols using network simulators. We used vehicular traces from the city centre of Luxembourg [47, 48], where we choose a particular area during rush hours. Our results show, first, that by using RSUs as back up nodes, more data objects are delivered in requester nodes, compared when all messages pass through RSUs. Second, by centralize the network, we can calculate efficient routing paths of messages and offload messages from the network, thus highlighting the efficiency of applying a centralized architecture in VANETs.

## 1.4 Thesis Outline

The remainder of the thesis is structured as follows.

Chapter 2 reviews the theoretical background and some of the related works that contributed to and were used for the approaches proposed in this thesis. Then, our main contributions are structured in two parts: Part I (Chapters 3, 4) introduces our work on Vehicle to Vehicle (V2V) communication and presents our two V2V proposed routing protocols. Part II (Chapters 5, 6) presents our proposed Vehicle to Infrastructure (V2I) techniques and protocols we develop in this thesis. In the following paragraphs, we summarize the contributions included in each Chapter.

Chapter 3 addresses **RQ1** by proposing our first two main routing protocols that deal with the continuous broadcasting of messages, as described in Section 1.3.1.1. The first is called Multipath, Multihop and Multichannel NDN Routing Protocol (MMM-VNDN). In MMM-VNDN we propose the use of MAC addresses as unique identifiers for each node, and we perform routing decisions based on these identifiers. In particular, we include new fields inside the NDN messages that contain MAC addresses. We broadcast every message and based on these addresses, each node decides if it will keep or discard the incoming message. Then, we select the next hop to transmit a message based on three different next hop selection techniques. The second routing algorithm that we propose extends MMM-VNDN and is called improved MMM-VNDN, iMMM-VNDN. In iMMM-VNDN we eliminate broadcast transmissions when it is possible. Thus, we transmit messages by selecting an appropriate MAC address from a node's routing table to unicast a message. Therefore, we leave the NDN messages intact without introducing new fields. Finally, as in MMM-VNDN, iMMM-VNDN also selects the next hop based on the same three different next hop selection techniques.

Chapter 4 addresses **RQ2** by focusing on directional forwarding of messages and creating paths in an NDN-VANET, as described in Section 1.3.1.2. We develop a new approach called enhanced Geographical aware routing protocol for NDN-VANETs, eGaRP. In eGaRP, each node is equipped with directional antennas. We introduce an antenna selection algorithm for transmitting a message only through one directional antenna. This helps to reduce the spreading area of the message. eGaRP's underlying protocol is iMMM-VNDN. This means that when one antenna is selected for directional forwarding towards a particular location, the message transmission from that antenna is unicast. Finally, with eGaRP, a node detects path breaks and/or congestion or collisions in the channel and, then, decides by itself when to send the message to another node. The detection of a broken path and the retransmission of messages are based on a timer-contention-based forwarding algorithm.

Chapter 5 addresses **RQ3** by introducing two new routing protocols for NDN-VANETs that use RSUs to route messages, as described in Section 1.3.2.1. In these protocols, we introduce two phases for forwarding a message. The first, called learning phase, is based on beacon transmissions between vehicles and between vehicles and RSUs. Through this beacon exchange, vehicles and RSUs know with which vehicles they are connected and populate their routing tables. Afterwards, we introduce our routing approaches. In the first, called *linked approach*, when a vehicle requests a content object, it always sends its request to the nearest RSU. Then, the RSU (based on the learning phase) transmits this request to the node that has the content object. Therefore, in the *linked approach*, RSUs act as a gateway between requesters and content sources. In the second routing algorithm, called hybrid approach, a vehicle decides according to its FIB table where to send a message. In case of a routing entry pointing to other vehicles, then the vehicle unicasts the message by selecting that entry. Otherwise, the vehicle sends its message to its nearest RSU. Therefore, in this approach, the RSUs assist in the routing of a message, only when a route to the content source does not exist and act as a backup mechanism to perform content retrieval.

Chapter 6 addresses **RQ4** by proposing a new routing protocol using SDN, as described in Section 1.3.2.2. Specifically, we study the impact of SDN on vehicular environments. To perform so, we use an SDN controller application (which is deployed away from city streets) to change characteristics of Road Side Units (RSUs) and to assist in message routing. The SDN controller has global knowledge of network connections. In addition, the SDN controller instructs vehicles on how to satisfy their requests, i.e. how and where to send a message, and finally, it notifies vehicles about emergencies, such as road accidents. We experiment with two different node configurations. In the first configuration, we install one omnidirectional and multiple directional antennas both in vehicles and RSUs. Vehicles are communicating with each other and with the deployed RSUs using the omnidirectional antenna via the control channel. Content retrieval is performed via the directional antennas in a service channel. In the second configuration, we install multiple omnidirectional antennas both in vehicles and RSUs. We treat these antennas as a MIMO system, and content retrieval is performed via this system. The SDN controller is connected to the RSUs and collects information about vehicular and network traffic from them. Then, the SDN controller calculates the number of connected cars to an RSU and decides whether to change its transmission power, if it calculates that more cars can connect to the RSU. Finally, the SDN controller having both local and global knowledge of the network topology, assists in path calculation, when a request is being issued and populates the routing tables of nodes on the path taking part in the content exchange process.

Finally, Chapter 7 concludes the thesis by summarizing the contributions of this work and addresses future work that can be investigated for dealing with the limitations of vehicular environments.

# 2

# Related Work and Theoretical Background

The following Chapter presents the key aspects of Vehicular Ad Hoc Networks (VANETs), Named Data Networking (NDN) and Software Defined Networking (SDN). This thesis is based on these three main concepts. In addition, this chapter presents the related state of the art work on these fields and their combination.

## 2.1 Vehicular Ad Hoc Networks

The emerging rise of mobile devices combined with advances in wireless communications over the last decade has led to the creation of Mobile Ad Hoc Networks (MANETs). These are mostly infrastructure less networks that are connected through a wireless signal. Vehicular Ad Hoc Networks (VANETs) are addressed in vehicles and are a sub-category of MANETs. Their creation is based upon the need for communication between vehicles in everyday life. A VANET architecture includes vehicles that are equipped with an On-Board-Unit (OBU) and infrastructure, such as Road Side Units

(RSUs) and/or cellular network infrastructure.

Nowadays, reliable communication between vehicles, together with the support of a variety of vehicular applications, is the key for success in vehicular networking. Vehicular applications are divided mostly into three categories: infotainment, traffic efficiency and management, and active road safety applications [85]. Infotainment applications include cooperative local services and global Internet services. Traffic safety applications include speed management and co-operative navigation, while safety applications include traffic information, road accident warnings and/or weather conditions [85]. For development and deployment of safety-related applications a VANET should gather, aggregate, validate and disseminate appropriate information with the specified requirements, e.g. low latency [77]. For instance, [100] develops a robust broadcast scheme for disseminating safety information in a VANET. On the other hand, [16] deals with the broadcast problem that exists in VANETs. This thesis deals with infotainment applications and how vehicles can retrieve the requested information.

For V2V communication and implementation of vehicular applications, the IEEE 802.11p standard has been proposed and is considered as the *de facto* standard [110]. For V2X communication, Cellular V2X (C-V2X) has been proposed. C-V2X uses the standardized 3GPP standardised 4G Long Term Evolution (LTE) or 5G cellular connectivity. Many studies have focused on the application of 4G and 5G to VANETs [14, 49, 68, 87, 114, 118] and how cloud systems can improve a VANET performance [23, 39, 41, 69]. Both standards operate in the 5.9 GHz band [73]. In this thesis, we use the Wireless Access Vehicular Environments (WAVE) protocol, which derives from the IEEE 802.11p standard. The WAVE protocol stack is composed of several components, most notable the 802.11p and allows a multichannel operation. The IEEE 1609.4 standard defines the operation of the WAVE protocol. It defines several channels, each for different application, and with different characteristics [70, 73].

As far as the communication scheme design in VANETs, many works have been proposed over the last 20 years. One of the most challenging characteristics in a VANET is the intermittent connectivity, derived from the vehicular movement and, therefore, from the vehicular path breaks. To assist connectivity, studies analyse the exact placement that the RSUs should have [149], while others use RSUs to assist for communication between vehicles [97]. Inter-vehicle communication is also used to gather traffic information amongst vehicles [129] to create platoon systems resulting in the enhancement of safety, traffic flow and road capacity [120].

## 2.2 Information Centric Networking

Information Centric Networking (ICN) is a future Internet architecture that aims to eliminate the concept of the host location that an object can be found. In particular, in ICN information is named at the network layer with unique identifiers and, then, is exchanged based on these unique identifiers [153].

A key challenge of ICN is how the routing of the desired information will be performed. There are many approaches. Some of them include flooding based approaches, where the request is broadcast to all of its connected nodes; intra-domain routing, where the request is routed based on distance vectors or link-state vectors. The main disadvantage of flooding is the excessive overhead that creates into the network by always broadcasting all packets. In comparison, distance vector and link-state routing algorithms suffer from slow convergence, infinite routing loops and limited scalability [86].

In general, ICN focuses on replacing the current host-based Internet architecture. Therefore, many ICN paradigms have been developed and are still in active development. The most popular of them are the following:

- Data Oriented Network Architecture (DONA [91]) is an ICN paradigm that replaces the URLs with standard names in each object. DONA supports on path caching for information availability.

- Publish Subscribe Internet Technologies (PURSUIT [63, 153]) is based on the publish-subscribe approach. In PURSUIT a source publishes information and a client subscribes to the content it needs. Again the names of the objects are unique and nodes can forward the content objects and cache them.

- COntent Mediator for content-aware nETworks (COMET [10]) designs mechanisms for mapping information names to particular servers and hosts depending on the state of the network.

- Named Data Networking (NDN [162]) will be described in detail in the rest of this Section.

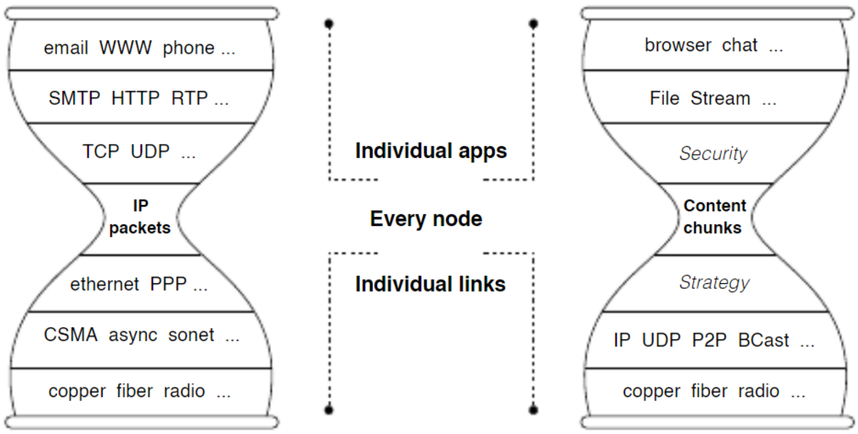



Figure 2.1: IP and NDN stacks. [162]

**Named Data Networking**

Content-Centric Networking (CCN) or Named Data Networking (NDN) [162] is a variant of Information Centric Networking (ICN) architecture. NDN is one of the most popular ICN paradigms and it is based on the principle that a content object can be distributed among network nodes only based on its name. In NDN the structure of a node changes, by eliminating the TCP/IP stack and replacing it by the NDN layer, where objects are being processed. The NDN stack is shown in Fig. 2.1. The objects in NDN are called content chunks and replace the IP packets.

Content-Centric Networking (CCN) or Named Data Networking (NDN) [162] is a variant of Information Centric Networking (ICN) architecture. NDN is one of the most popular ICN paradigms, and it is based on the principle that a content object can be distributed among network nodes only based on its name. In NDN the structure of a node changes, by eliminating the TCP/IP stack and replacing it by the NDN layer, where objects are being processed. The NDN stack is shown in Fig. 2.1. The objects in NDN are called content chunks and replace the IP packets.

NDN messages are either requests, named Interest messages, or contain the information requested, named Data messages. Hence, when a node requests a content object it sends an Interest message and when a node responds to this Interest message with the requested content object, it sends a Data message. Each node in NDN has a unique structure that is based on particular Data structures. The Content Store acts as a cache and stores content objects. The Forward Information Base (FIB) table acts as a routing table. The Pending Interest Table (PIT) keeps track of the

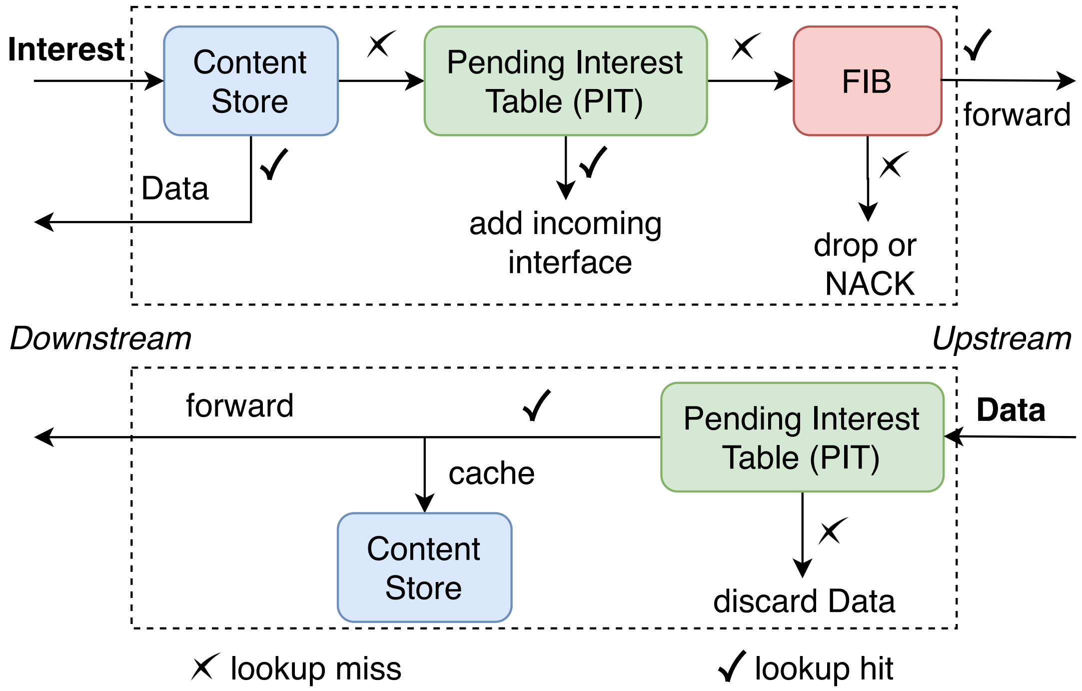



Figure 2.2: Interest processing at an NDN node. [162]

forwarded Interests to avoid retransmission of the same Interest.

Fig. 2.2 describes the process of the Interest message when it arrives at a node. First, the node checks its Content Store Table to check if the content object is cached. If it is, the Data message is retrieved and is sent back to the node that sent the Interest. If the Content Store does not contain the requested content object, then the node compares the Interest name to its PIT. If there exists a match, this means that the node has forwarded the Interest before and there is no need for further transmission. In this case, the node stores the interface where the Interest came from in the PIT and then discards the Interest. If there is no match, the node inserts the Interest together with the interface that the Interest came from into the PIT. Then, the node checks its FIB table to identify possible interfaces to transmit the interest. The FIB acts as a routing table. The Data message follows the PIT entries to be forwarded back to the requester node.

## 2.3 Applying ICN in Vehicular Networks

The main problem of NDN lies in its application to large-scale networks. Firstly, today's technology (both from a hardware and software point of view) is not ready to support

NDN in an Internet scale architecture [116]. Named Data Networking was designed for fixed networks, where each server has many interfaces to communicate with the connected to it nodes and can select one interface to send and/or receive Interests and/or Data messages. Many works apply the NDN principle in fixed networks [32, 99, 108, 140, 156] and exploit multiple paths for Interest forwarding [52, 122, 163]. For instance, [144] proposes to split the Interest messages of the same request and send them through multiple paths at the same time. Furthermore, study schemes for congestion control in NDN static networks [40, 113, 160] have been proposed. [53, 113] propose congestion control mechanisms, by utilizing the Round Trip Time (RTT) estimations to measure the available bandwidth. Then, these works send Interests though the best face based on estimations (for example bandwidth estimation), and by implementing the AIMD congestion control mechanism.

Studies also focus on applying the NDN paradigm to Mobile Ad Hoc Networks (MANETs). [96] uses a controller that controls each routing strategy of every mobile device. It uses multiple paths to forward Interests towards specific servers. In [17] a geographic Interest forwarding scheme is proposed, where the goal is to balance the energy consumption in the network. In [30, 31] users are being selected to transfer popular content objects.

Moreover, in wireless communications all NDN routers broadcast Interest packets to neighbour nodes to receive the requested content. Based on this assumption, in a large-scale network the size of FIB tables on each NDN node is increased due to the high number of possible connections and content sources. Subsequently, the possibility of congestion at paths through the network is very high, the size of the FIB table of each node is large, and thus, the computational time of a simple retrieval process is increased. In that case, in a mobile network like a VANET, adequate methods for updating and deleting the FIB tables of all nodes should be developed.

There have been several attempts to encounter these problems; some of them try to identify a path between a requester and a content source and store this path for future references [29]. Authors in [26] propose CCVN, content Zcentric networking for VANETs, where they extend the framework of CCN to support content delivery on top of IEEE 802.11p. They broadcast messages in the entire network, but in case of collisions, the messages are re-broadcast according to the distance from previous senders. In [156] authors suggest using the different interfaces of a CCN node and transmitting Interests simultaneously through multiple interfaces. In addition, they propose the utilization of NACK messages to avoid the dissatisfaction of an Interest.

Another major concern is the content distribution in an NDN-VANET. In a VANET, broadcasting requests is a common solution for retrieving data. But, since broadcasting consumes many network resources, not all requests should be broadcast. A solution for this is to classify broadcast protocols with the premises that not all broadcasts are the same. Each protocol should be context-aware based on the application requirements [55]. Furthermore, content prefetching poses as a major research issue, as it allows the increase of content availability throughout the network. Content can be prefetched in static network nodes throughout the path that a vehicle is travelling [107]. Also, content can be cached in RSUs that are placed along the road such as to eliminate redundant caching, i.e. for the content to be available everywhere without wasting caching resources [15]. Moreover, NDN can be extended with publish/subscribe approaches to provide efficient data collection and dissemination [56]. Vehicles can also exchange content information about their cached contents during their communication process [158]. In the following subchapters, we present four relevant works that use NDN in VANETs.

### 2.3.1 Content-Centric Networking in VANETs (CRoWN)

One of the first studies combining CCN with VANETs is the Content-Centric Networking in Vehicular Ad Hoc Networks (CRoWN) [27]. CroWN is a scalable network protocol that controls channel overhearing. It also employs multihop forwarding of messages and Data caching to support node mobility.

In the CCN/NDN concept, the content object that is requested is fragmented in chunks. This avoids overflowing the channel by exceeding possible bandwidth limitations that are posed from the dedicated WAVE standard (c.f. Section 2.1) and to avoid possible congestion. Subsequently, for a node to receive successfully a complete content object, it should send as many Interest messages as the number of content chunks that the content object is divided into.

CROWN employs the previous statement. A node requesting a content object sends the first Interest message, named Basic Interest (*BInt*), to request the first content chunk and to start the retrieval process. The following Interests, named Acknowledgement-Interests (*AInt*), are used to request consequent content chunks and to acknowledge previously successfully received Data messages.

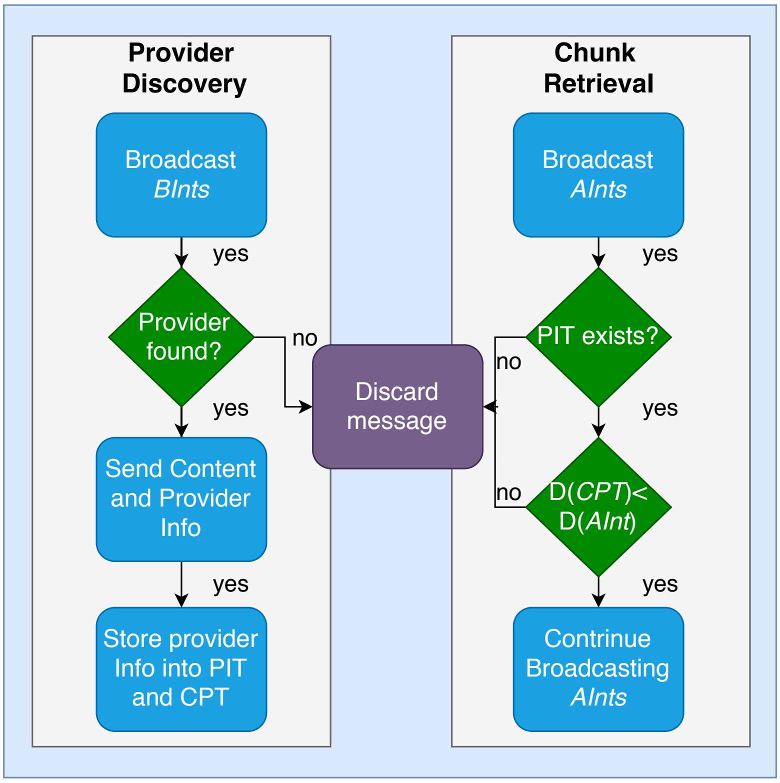



Figure 2.3: CRoWN discovery and forwarding stages

The protocol consists of three data structures. As in native CCN, a node in CRoWN consists of a Pending Interest Table (PIT) for keeping track of forwarded Interest and a Content Store to cache received Data. Moreover, it employs a new data structure named Content Provider Table (CPT) to keep track of the available content providers.

In CRoWN all message transmissions are broadcast. The scheme functions in two stages. The first stage is named Provider Discovery. In this stage, nodes broadcast *BInts* to discover available content providers around them. If a node receives a *BInt* and does not have a content in its cache (or does not provide a content), then, it continues broadcasting the *BInt* to its neighbours. To reduce redundancy and broadcast storms, nodes overhear *BInt* transmissions. When a *BInt* transmission is overheard, nodes cancel their own scheduled transmission of the *BInt*. When a node receives a Data message, it caches it into its Content Store, it maintains its entry into the PIT and it inserts an entry into its CPT.

The second stage of CRoWN is the chunk retrieval, where requester nodes broadcast *AInts* into the network. Inside the *AInt* a requester adds the distance from the selected provider, as well as an acknowledgement of previous received Data messages. An intermediate node that receives an *AInt* will continue broadcasting it only if there is an entry into its PIT and if it realizes that it is closer to the selected provider than the field into the *AInt*. Both stages are shown in Fig. 2.3.

The main disadvantage of CRoWN is that all requests are always broadcasts. Broadcasting leads to packet collisions in the channel, unnecessary utilization of network resources and can create broadcast storms leading to lower network throughout.

### 2.3.2 Controlled Data Packets Propagation in Vehicular Named Data Networks (CONET)

In VANETs broadcasting every message is considered the standard way of communication, because of the broadcast nature of the wireless medium. Therefore, in an NDN VANET, when a node requests a content object, it broadcasts an Interest message into the network. The corresponding Data message follows the PIT entries of the Interest, but it is still broadcast into the network. The Data message is discarded if a node has not requested this content that the Data describes.

Hence, to deal with the broadcasting of a Data, CONET (Controlled Data Packets Propagation in Vehicular Named Data Networks) [20] uses a hop-count in each node that the Interest message passes. Every time the Interest is transmitted to a node, this hop count is incremented by 1. When the Interest reaches a node that can respond with the corresponding Data, the hop count of the Interest is extracted and it is added to the field of the Data message named: "TTL" (Time To Live). Then, the Data is broadcast to the network. Every node receiving the Data will decrease the TTL field by 1. If a node receives the Data message that its TTL field is 0 or less, then the Data is discarded. By this, CONET eliminates possible duplicate transmissions of the Data, leading to lower potential congestion and fewer broadcast storms.

CONET deals with Data propagation without minimizing redundant Interest transmissions. CONET deals with the broadcasting of a Data message by transmitting a Data message, when the hop counter is higher than zero. But, since the Data message follows the entries on the PIT of an Interest message, this Data message will be propagated to multiple nodes before this hop counter expires, creating unnecessary transmissions and utilization of network resources.

### 2.3.3 Density-Aware Delay-Tolerant Interest Forwarding Strategy in VANETs (DADT)

To deal with the several issues that the Interest broadcasting in NDN in vehicular environments creates, [95] proposes a Density-Aware Delay-Tolerant (DADT) Interest forwarding strategy to retrieve traffic data in vehicular NDN. Interest broadcasts create broadcast storms, which result in much packet loss and huge transmission overhead. DADT is a solution for this by using geographical characteristics to retrieve traffic data using Delay Tolerant Networking.

In DADT, every node is equipped with a GPS device to know its current coordinates at any given time. DADT forwarding includes two communication phases: rebroadcast and retransmission. In the rebroadcast phase, each node receiving the Interest creates a timer and forwards the Interest after this timer expires. Hence, DADT defines a forwarding priority based on the timer on each node. The factors that affect this priority are:

- The nodes farther away from the last hop should have higher forwarding priority so that the Interest can be propagated faster with fewer hops.
- The nodes closer to data location should have higher forwarding priority, thus, the Interest could have more chances to be forwarded to its desired area.

When a node waiting to transmit an Interest message overhears the Interest transmission, it stops rebroadcasting of the Interest and deletes it.

In the retransmission phase, DADT defines a spatial priority, which measures how close the neighbour is to the shortest line between the current node and the data location. This priority is calculated according to the neighbour list. Interests will be retransmitted only if the priority is higher than zero. Before retransmitting an Interest, if a node overhears the transmission of the same Interest from another node, it will cancel its own Interest retransmission.

DADT applies selective broadcasting, meaning that it reduces the number of broadcasts transmissions but does not minimize them. Broadcasting increases the network traffic, wastes resources and lowers the bandwidth of the network.

### 2.3.4 Geographical Opportunistic Forwarding Protocol in VANETs (GOFP)

To support path breaks and deal with the location-independent routing that exists in traditional NDN [98] proposes a Geographical Opportunistic Forwarding Protocol (GOFP) for vehicular networks using NDN. GOFP is based on the fact that constant connectivity between the vehicles and the service platform cannot be considered reliable because of sparse vehicle density or high mobility of vehicles. GOFP also supports the store-carry-and-forward paradigm, meaning that a vehicle can store an Interest as in NDN, carry an Interest, instead of instant forwarding it as in native NDN, and forward it according to the routing strategy.

To perform such a task, GOFP incorporates the geographic location of a particular location-dependent application (such as parking space) named Point of Interest (POI) into the naming of an Interest. Moreover, in GOFP the messages (both Interests and Data) contain a Time-To-Live (TTL) field that defines the lifetime of the messages. These messages are discarded once the TTL expires. Finally, messages contain a trajectory info field that shows the consumer's (in case of an Interest) or the content provider's (in case of Data) trajectory. A node requesting a content object will send the Interest to a neighbour vehicle only if the neighbour vehicle moves closer to the POI than the content requester.

When a vehicle that has the content object receives the Interest message, it will forward the Data either to a vehicle that moves towards the content provider or to a vehicle that has higher speed (and moves towards the content provider) than it.

GOFP proposes geographic forwarding of messages, which is considered as a viable solution for vehicular networks. But still, NDN principles are compromised if Data do not follow the PIT entries of the Interest, but they are only forwarded based on the location of the content requester nodes.

### 2.3.5 Interest Forwarding Based on GeoLocations in VANETs (Navigo)

One of the main principles of NDN is the decoupling of content objects from their locations. A node can request a content object just by knowing its unique identifier. This is beneficial for VANETs that are characterized by high mobility and path breaks.

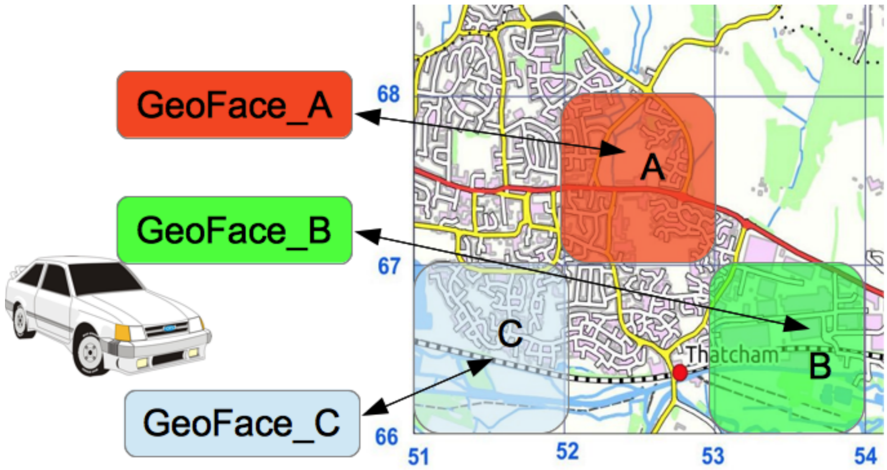



Figure 2.4: Mapping Geofaces to particular areas [72]

In such a network, the standardized way of retrieving a content object is to broadcast requests to neighbour nodes, expecting that one node will respond with the requested Data.

Navigo [72] exploits NDN. To reduce overhead, Navigo introduces geocasting strategies for content retrieval. One of the main challenges in the combination of NDN with VANETs is how to identify properly content providers. In this case, Navigo introduces geofaces that bind a particular content object to a particular location on the map on the premises that vehicular applications usually concern a particular geographical area. Specifically, in Navigo, the world is divided into regions according to the Military Grid Reference System (MGRS), c.f. Fig. 2.4.

When a node sends an Interest message, it broadcasts the Interest towards all locations. A node responding to this Interest will include in its Data its geolocation. The requester and all intermediate nodes receiving the Data will bind the Data to this location on the map for future requests. The routing strategy of an Interest is, therefore, the following:

- A requester searches its FIB to identify possible entries to forward the Interest. If such entries exist, then the Interest is forwarded towards the geolocation contained in the FIB entry. If there are not, the Interest is broadcast. The latter is called exploration phase.
- If multiple entries are available in the FIB, the Interests are routed in a round-robin way. If only one FIB entry is available, the Interest is forwarded to this

geolocation with a probability $p$. The Interest is broadcast with a probability $1 - p$.

- If the Interest times out, then the FIB entry (and therefore, the geolocation) is removed and another FIB entry is selected. If there is not another FIB entry, then the Interest falls back to the exploration phase.

Furthermore, to forward an Interest towards a particular geolocation, each node calculates a path towards the destination. This path is calculated using the Dijkstra shortest path algorithm by assigning each street as an edge and an intersection as a node. Then, particular costs are assigned to edges that are inversely proportional to the number of lanes. When an Interest is incoming in a node, the node forwards the Interest only if it is closer to the destination area (destination geolocation) than the node that sent the Interest.

Navigo uses geographical locations to forward Interest and Data messages. Navigo helps by reducing the number of retransmitted Interest packets and, hence, it avoids Interest flooding. However, in Navigo, content names include the location of the content provider. This limits the use of in-network caching and causes major changes in the FIB table structure. The change of native NDN architecture is not preferable, especially for large-scale deployment [88].

### 2.3.6 Multiple Unicast Paths Forwarding Protocol (MUPF)

As mentioned in Section 1.2.1 packet broadcasting has a significant impact in VANETs, since broadcasting leads to packet loss. To reduce broadcasting, [123] proposes a multiple unicast paths forwarding (MUPF) protocol that makes the Data packets return along opposite paths, efficiently decreasing the excess of useless network traffic. MUPF uses motion parameters of routing nodes and link quality metrics, such as link expiration time and link available probability, to choose the next hop of a message.

MUPF suggests the creation of two new messages named REQ and REPLY. When a content requester wants to send an Interest, it checks its FIB to identify next hops to send the message. When there are no entries into its FIB, the node broadcasts a REQ message. Every node receiving the REQ message replies with a REPLY message that is also broadcast into the network. If a node is a content provider, its id is added to the REPLY message.

MUPF adds a Weight Table (WT) and Neighbour Table (NT) in each node to record the motion parameters and neighbour information. The motion parameters are:

1. the distance that the two vehicles have,
2. the direction that denotes the direction of the next hop,
3. the traffic density, which is defined as a function of the road connectivity to the average number of neighbours within a given transmission range per second.

Based on WT and NT, MUPF creates unicast paths from the requester node to the content provider. These paths are inserted into the FIB table of the requester node. When multiple content providers exist, multiple entries to different providers exist in the requester's FIB. In this case, a content requester uses link quality metrics to select the path that the Interest should go through. Fig. 2.5 shows the forwarding of the Interest and a Data message according to MUPF.

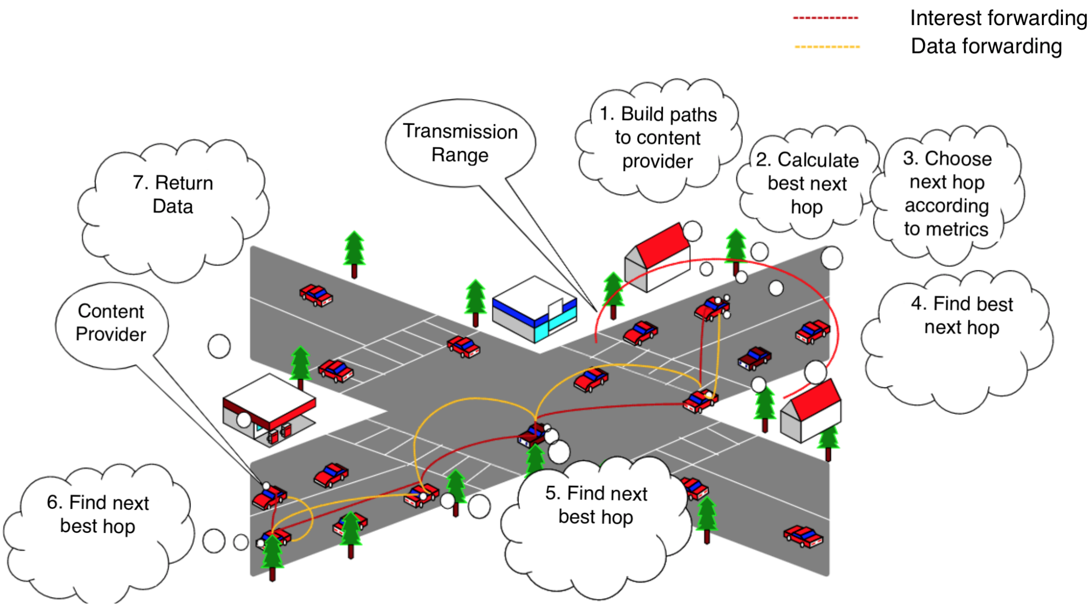



Figure 2.5: MUPF Interest and Data forwarding [51]

MUPF creates new messages for discovering neighbours and content providers. These new messages create excessive traffic for the network and are not compliant with the wireless vehicular standards and especially the WAVE protocol. Moreover, MUPF deals with path breaks by broadcasting REQ and REPLY messages, each time a path between a content provider and a requester breaks. This also creates unnecessary traffic to the network. Finally, MUPF fails to deal with path breaks, when the Data message is forwarded back to the requester node.

### 2.3.7 Hybrid Forwarding Strategy Using NDN in VANETs (HVNDN)

To avoid always broadcasting message and the lack of routing entries in FIB tables of nodes, [51] proposes HVNDN, a hybrid forwarding strategy through NDN. Specifically, HVNDN introduces an opportunistic and probabilistic forwarding strategy based on the geographical location of nodes. In particular, applications and routing entries are characterized by their location dependence or independence and Interests are forwarded based on this information. Moreover, HVNDN uses a new retransmission and acknowledgement mechanism to guarantee packet reliability.

In HVNDN all vehicles are equipped with two Data structures, the PIT and CS. There are two types of Interests, Location Dependent Interests (L-Int) and Blind Interests (Location Independent) named B-Int. Each vehicle contains a unique id. HVNDN assumes that the id and the location of the content provider are known. When an application is location dependent, a content requester broadcasts an L-Int by indicating the geographical coordinates of the Interest location. This Interest contains the geographical coordinates of the content requester and the destination coordinates of the content provider. When an intermediate node receives the Interest, it will check whether the node is closer to the content provider, by checking its own coordinates and the coordinates of the Interest. If the node is not closer to the content provider, then the Interest will be discarded. Otherwise, the node will check its CS and PIT to identify if it has the requested Data cached, or it has forwarded the Interest before. If these checks fail, the intermediate node defines a timer and broadcasts the Interest when this timer expires. In addition, if a node does not receive an acknowledgement before the expiration of this timer, it will also retransmit the Interest.

Nodes responding with a Data message will delay its transmission to avoid collisions. Data messages contain an acknowledgement field that is being set when nodes forward the Data. To avoid the waste of bandwidth, nodes broadcasting the Data set the value of the acknowledgement field to valid to inform vehicles around them to cancel their forwarding of the same Data message.

When an application is location independent, a content request broadcasts a B-Int message. All nodes receiving this B-Int message will continue forwarding based on a probability. This probability depends on the velocity of the vehicle, by its distance from the content source and its transmission range. The Data are being forwarded according to NDN mechanisms.

Using acknowledgements in vehicular networks is suboptimal because paths between vehicles can break unexpectedly. If a path breaks during the Data propagation from the content provider to the content requester, intermediate nodes that the Interest passed through will retransmit the Interest. If an intermediate node has moved away from the content provider, this leads to excessive Interests retransmissions that will burden the channel with unnecessary packets.

### 2.3.8 IP-Based Vehicular Content-Centric Networking (IVCCN)

In vehicular CCN networks broadcasting messages can increase content retrieval costs, since it usually creates redundant copies in the network, introduces high overhead, and wastes network resources. In addition, the native nature of CCN is for the Data message to follow the PITs that the Interest passed to go back to its destination. The latter, though, can create partitioning in a mobile network, when a node leaves and breaks the routing path. To solve these problems, IP-based Vehicular Content-Centric Networking (IVCCN) [148] is presented. IVCCN reduces the content object acquisition cost and improves the success rate by using the address-centric unicast instead of the content-centric broadcast transmission for content object acquisition.

In IVCCN the content object is related to a given geographical location, and the content retrieval is based on message exchanges between vehicles and Access Points (APs). In general, an area is divided into logical rectangular subnets. The subnets are placed next to each other so that all areas are covered. Routing between subnets can be performed. Since the subnets are not overlapping, one unique identifier can be assigned to each subnet. Fig. 2.6 illustrates the proposed network architecture consisting of vehicles, access points and access routers (AR) that connect many APs and are connected to the Internet.

IVCCN defines unique unicast addresses and content addresses. This distinction is based on the fact that a unicast address is used for routing a content request, without flooding the network with broadcast transmissions. A content address is used for seeking the content object in a given geographical location. In general, the unicast address and the content address define the source address and the destination address

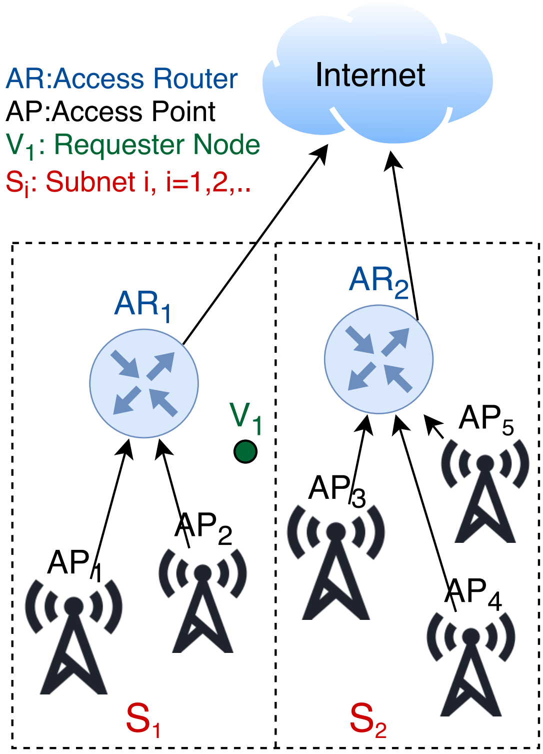



Figure 2.6: IVCCN architecture

of a message (not necessarily in this order).

A vehicle moving into different subnets constructs a unicast address independently, depending on which subnet it is located in. When a vehicle produces a content object and enters a new subnet, it advertises this content object to the AP that is mapped to this area. In particular, the content provider creates a message, named *shared* message, with the following fields:

- The source address is a newly constructed unicast address. The unicast address contains the geographical coordinates and the unique id of the provider.
- The destination address is a newly constructed content address. The content address contains the coordinates of the AP, the coordinates of the content object (in this case these are the same as the coordinates of the content provider) and the content object id.
- The payload which includes the content object.

Then, the vehicle sends this *shared* message, and this message is forwarded hop by hop until it reaches the AP. Every intermediate vehicle receiving the *share* message adds the content id, the content provider geographical address, and the payload in their content table.

Let us assume that a vehicle requesting a content object is located inside a subnet with an AP. We also assume that this content object is related to a particular area, and the requester knows the geographical coordinates of the content object. Finally, we assume that the AP is located closer to the provider than to the requester. For instance, in Fig. 2.6 assuming that the content provider is located south of $AP_2$, $V_1$ is located closer to $AP_2$ than to the content provider. In this case, the vehicle requesting the content object, first, sends a *C-Req* message. The *C-Req* consists of one unicast and one content address:

- The unicast address contains the geographical coordinates and the id of the requester and is used as a source address.
- The content address contains the geographical coordinates of the AP that is located in the current subnet and the content object address and is used as a destination address.

When an intermediate vehicle receives the *C-Req*, it sends back the requested content object, if it possesses it. Otherwise, the *C-Req* reaches the AP. If the AP has the requested content object, the AP sends it back to the requester node. If the AP does not have the requested content object, it initiates a content creation algorithm to obtain the requested content object. The content creation algorithm is the following:

- First, the AP creates a *create* message, without changing the source and destination addresses.
- Second, the AP sends this *create* message towards the content provider.
- Third, the AP receives the content object (either from an intermediate node or from the content provider) via a *share* message.

After the successful delivery of the content object to the AP, the AP sends the content object back to the requester node.

If we assume that the content provider and the content requester are located in different subnets, then the content requester will send the *C-Req* message to the AP that is located in its subnet. After this AP will forward the message to its connected AR. The latter will send the message to the AR that is connected with an AP inside the provider's subnet. Then, the message will be forwarded to the AP located inside the content provider's subnet and, finally, this AP will forward the message to the content provider. The content object will follow the same route to return to the requester node while being cached in every intermediate node that passes by. For instance, in Fig. 2.6, if the content is located in $S_2$, then the requester $V_1$ sends its request to $AP_2$. $AP_2$ sends it to $AR_1$, $AR_1$ sends it $AR_2$ and the latter to the AP located closer to the provider.

## 2.4 Software Defined Networking

In this Subchapter we describe Software Defined Networking (SDN) and we present studies that combine SDN with NDN.

### 2.4.1 SDN Architecture

Software Defined Networking (SDN) is a paradigm for network architectures with the potential to reshape the field of computer networking [92]. SDN's key principle is programmable networks that are centralized and flexible. SDN considers increasing network functionality by separating the data and the control plane as well as by logically centralizing network control. SDN simplifies networking in both development and deployment of new protocols and applications. In SDN, the control plane controls the network infrastructure and installs forwarding rules to it. Network infrastructure just accepts all the actions from the control plane without implementing and running its own software and protocols.

The generalized simplified SDN architecture [93] can be seen in Fig. 2.7. The architecture consists of three layers:

- The application layer, where all applications are running.
- The control layer, where all applications are programmed. The control layer provides supervision of the network forwarding state and makes the network application abstract.
- The infrastructure layer that consists of all the network components like routers, switches etc. Switches are controlled through software running in an external,

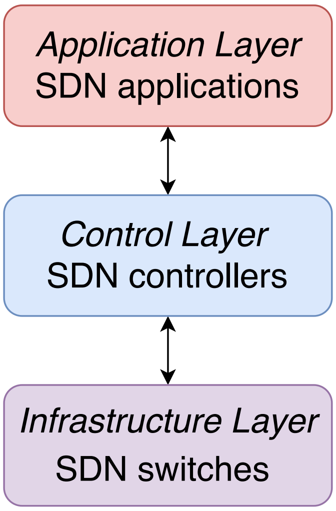



Figure 2.7: SDN layers

decoupled control plane [76].

SDN separates the control from the data plane through the control and the data plane abstraction. The data plane consists of forwarding elements such as virtual and physical switches. These switches allow for packet switching and forwarding. SDN includes a flow table for packet forwarding that contains flow configurations and matches each flow with the action named in the table. The control plane provides a global overview of the network. This global overview provides a collection of network nodes. The nodes' forwarding tables support features that link information, queues and ports and attribute the node's switching potentiality [76].

Some efforts have been made to bring the concept of SDN into wireless networks [58, 127, 128, 136, 165]. However, applying SDN to the wireless domain is challenging due to the fact that wireless networks feature many characteristics that rarely exist in wired networks. Because of the intrinsic nature of non-stationary wireless channels, the wireless communication service quality is subject to wireless transmission power, interferences, etc. [79]. To tackle these problems, [126] creates a testbed management system to evaluate various types of SDN applications in wireless sensor networks. SDN has been proposed to cope with the challenges of mobile networks, such as VANETs. SDN in VANETs has the potential to offer flexibility, programmability, centralization, elasticity and agility [101]. [142, 155] describe an architecture combining SDN with Fog computing for management and orchestration of vehicular networks.

Moreover, SDN can be combined with ICN to offer flexibility and scalability in

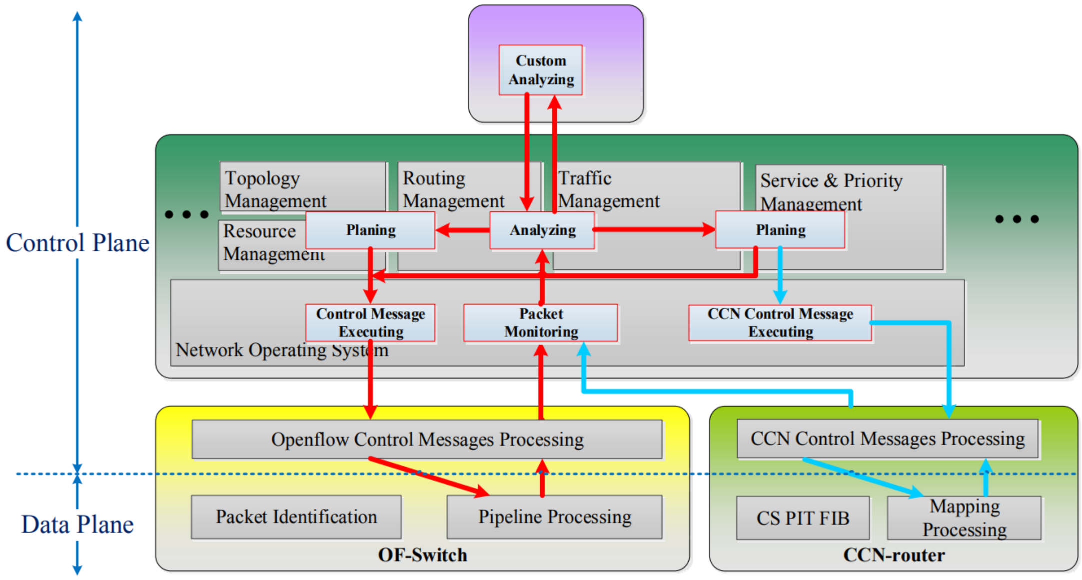



Figure 2.8: SDN-based CCN traffic management architecture [137]

networks [102]. In wireless communications, ICN and in particular, CCN/NDN native broadcasting of Interest messages for content discovery create usually unnecessary network redundancy, something that SDN could limit. By combining SDN with NDN and by assigning one local SDN controller to a cluster of nodes, [135] creates a forwarding strategy that allows for robust communication between nodes. An extension of CONET (Section 2.3.2) has also been proposed [145]. In the next chapters, we describe some core studies that combine ICN with SDN.

### 2.4.2 SDN-Based CCN Traffic Management

An SDN-based autonomic CCN traffic management is presented in [137]. The network architecture is presented in Fig. 2.8. It consists of an SDN controller, SDN switches and CCN routers. The switches and the routers cooperate and perform specialized actions.

In the SDN switch, the CCN protocol is installed. The data flow processing of a switch is divided into four phases:

- Flow identification identifies a CCN flow.
- Processing Type Classification recognizes the processing type and then goes

into different pipeline processes.

- Pipeline Processing writes the content object name into a flow entry.
- Queue Scheduling, where users indicate which packets will be sent to which queues.

To enable CCN in SDN, the existing data structures as defined in Section 2.2 are required in the CCN routers. But, to enable CCN only having the same data structures is not sufficient; additional functions are developed to enable the centralization that SDN offers. First, a communication function is performed, where the routers communicate directly with the SDN controller or with one of their neighbours. The second function is the traffic configuration, where CCN routers add functional fields to packets such as queue scheduling and queue management.

Finally, the SDN controller is connected with all other network devices and is in charge of configuring all rules for packet processing. In addition, the controller manages the routing, since it receives messages both from switches and routers and can calculate (according to the desired routing protocol) paths. Then, the controller creates flow rules and sends them to the switches to be executed. Last, the controller receives packets from all network components and uses them for different functions, such as classification of network nodes, scheduling rules, etc.

[137] does not perform any simulations to evaluate their proposals. Hence, it is unclear if the proposed architecture would burden the network and is a feasible solution for network management.

### 2.4.3 SDN-Based Routing Scheme for CCN (SRSC)

The SDN-based routing scheme for CCN(SRSC) architecture is depicted in Fig. 2.9. SRSC uses only the typical CCN data structures and Interests and Data messages for communication between routers and controllers, as described in Section 2.2. When an Interest is issued from a requester (in this study a requester is a node containing a router), if there is not an entry into the FIB table, the Interest is being sent into the SDN controller for path calculation. The SDN controller has a global overview of all network connections. To obtain this overview, the controller performs a bootstrapping step: The controller broadcasts an Interest message to the network. Nodes receiving this Interest message respond to it with their id. Then the forwarding step is executed:

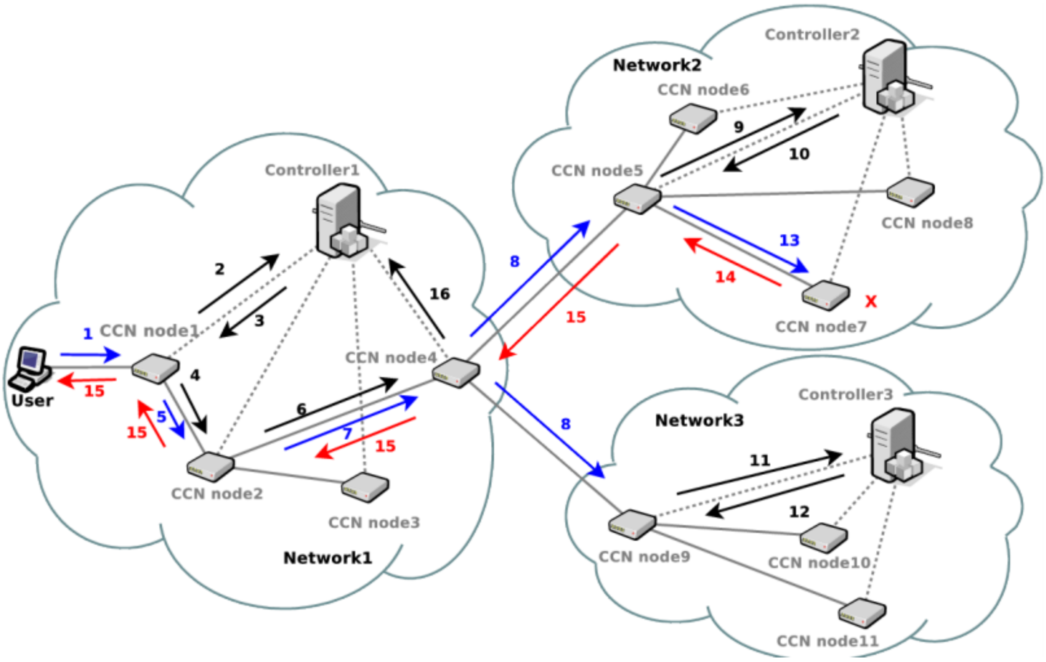



Figure 2.9: SRSC proposed architecture [35]

The router issues an Interest, and in case of no match in its FIB or Content Store, the Interest is being sent in the SDN controller. Then, the controller finds a path to satisfy the Interest and sends this path to the requester node. The requester node extracts the path and sends the Interest to the node according to the calculated path.

SRSC does not consider the traffic on each node. Hence, when a node is connected with many, all network traffic can be passed through that node, leading to congestion, collisions and eventually the node can start rejecting all incoming to it messages.

### 2.4.4 Software Defined Content-Centric Network (SDCCN)

In the Software Defined Content-Centric Network (SDCCN) the CCN architecture is combined with the SDN paradigm for content retrieval [43]. The network topology consists of CCN routers, users, and a CCN controller. The routers have the same structure as in Section 2.2. SDCCN adds one additional data structure named: Cache Rules Table (CRT). CRT stores rules for the successful caching or the rejection of a content object.

In SDCCN, when an Interest is requested, if there is no match to the data structures, it is forwarded towards the CCN controller. Then, the controller is responsible for installing forwarding rules to the FIB table of the requester. In addition, when a Content Store of a node overflows, the node will send a control message to the controller notifying about the lack of storage. The controller installs storing rules to the CRT and can also remove content objects from the congested Content Store.

In SDCCN, the controller does not create paths according to the current network state. In addition, a new data structure is proposed, something that compromises the CCN/NDN principle.

### 2.4.5 Use Cases of Applying SDN and NDN to VANETs

Use cases that improve the performance of the overall VANET, by using Named Data Networking (NDN) in vehicular nodes together with Software Defined Network are presented in [79]. Road Side Units (RSUs) are deployed in city streets and act as switches. These RSUs communicate with controllers that act as the control plane of the network. The use cases of combining SDN and NDN in VANETs are the following:

- Population of Forward Information Base (FIB) tables in nodes via an SDN controller. The SDN controller having an overview of all network links can populate the FIB tables of vehicular nodes to facilitate routing. This use case is implemented in Chapter 6.

- The SDN controller can dictate to vehicles with multiple interfaces installed which interface is proper to use. This selection is based on the connected links through these interfaces and is dependent on the content objects that vehicles exchange. This use case is implemented in Chapter 6.

- Exchange of messages between different region inside a city. Specifically, inside a city, we define clusters of vehicles. These clusters are formed based on the coverage area of an RSU, i.e. all vehicles existing in the coverage area of an RSU can form a cluster. Then we utilize SDN to dictate to vehicles moving between different clusters to transfer potential messages. This is depicted in Fig. 2.10.

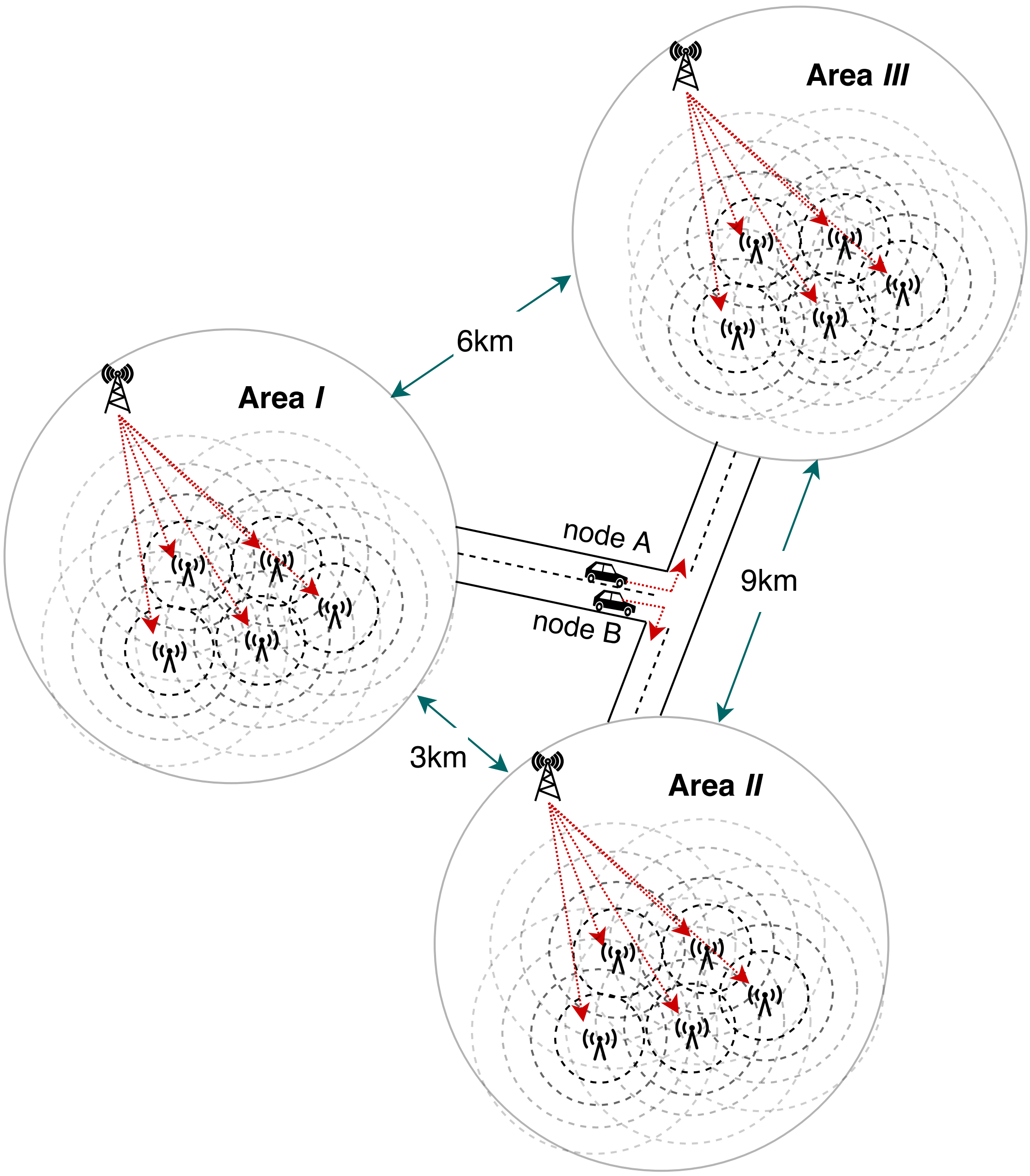



Figure 2.10: Communication between different areas that are defined by RSUs range

### 2.4.6 Conclusions

In this Chapter we describe core principles and characteristics of VANETs and NDN and present studies that utilize NDN in VANETs. Moreover, we describe SDN and cite studies that combine SDN and ICN in fixed networks. Although many works that integrate ICN in VANETs have been proposed, the issue of broadcasting every message is still open. Many works do not consider the issue of Interest broadcasting (c.f. Chapters 2.3.1- 2.3.2), others propose to create counters to terminate message broadcasting when the counter exceeds a threshold (c.f. Chapter 2.3.2), other propose to reduce the number of broadcast transmissions using DTN (c.f. Chapter 2.3.3), to use geographical locations to route a message (c.f. Chapters 2.3.4-2.3.5) or to create new broadcast messages to deal with path breaks in VANETs (c.f. Chapters 2.3.6-2.3.7]). But, still the open issue remains on how we can reduce or even eliminate the broadcast transmissions. We address this issue on Chapter 3, by unicasting both Interest and Data messages in a VANET.

Moreover, ICN integration in VANETs does not address the open issue of how to limit the dissemination area of transmitted messages. Studies have proposed the use of directional antennas and beamforming techniques ( [57, 61, 151, 152, 154, 161]) in VANETs, but without evaluating how it affects application performance. Therefore, in Chapter 4 we use directional antennas to target the destination nodes together with NDN and message unicasting, and we show that this improves the performance of the NDN application.

Infrastructure can address the communication limitations that exist in V2V communication. For instance, a large number of interconnected devices (vehicles) permanently broadcasting requests and establishing connections among them can lead to collisions on the channel and to message errors. Using infrastructure the impact of such problems could be reduced. Moreover, RSUs can assist in message transmissions, path selections or timer constructions for transmitting messages [138]. But still, the open issue remains how using RSUs as a main network component can affect the performance of NDN applications installed in vehicles. Therefore, in Chapter 5 we propose the integration of ICN in VANETs with RSUs that act as a main network component. RSUs act as either a central gateway, i.e. every message passes through an RSU, or back-up gateway, where every node decides whether a message should be sent towards an RSU.

Although, SDN has proposed for wireless networks [58, 127, 128, 136, 165], its applicability is still an open issue. On the other hand, SDN can deal with the permanent broadcasting of messages in an NDN-VANET. But still the challenge that is not answered in the presented studies is whether the integration of SDN in NDN-VANETs would be beneficial. Specifically, how could SDN improve the NDN application performance and how this impacts the network. The network conditions would be significantly different, for instance, when a consumer node sends 100 Interests and it receives 100 Data responses, than when a consumer nodes sends 100 Interests, receives 50 Data responses and periodically retransmits the unsatisfied Interests. Therefore, in Chapter 6 we use SDN in an NDN-VANET and we study its applicability and its impact in the NDN application.

# Part I

# Vehicle to Vehicle Communication

In this part routing algorithms using NDN for Vehicle to Vehicle (V2V) communication are presented. The main idea for V2V communication is that a vehicle should be autonomous and decide by itself, when, where, and how its content requests should be sent. In particular, in Chapter 3 we focus on reducing message broadcasting transmissions and we create paths between vehicles. To perform such a task, requests from vehicles are being broadcast periodically to discover neighbour nodes. During broadcasting, routing entries between vehicles are being created and inserted into the routing tables. After creating paths, when a vehicle produces or receives a request from another vehicle it unicasts this request to another node based on its routing table. Hence, we use paths consisting of many nodes to create multihop connections between a requester node and a provider node by unicasting messages. In Chapter 4 we focus on limit the dissemination area of a message. To perform this, we install directional antennas in vehicles, and each vehicle decides when, how and where a message should be sent. Thus, in this part, we focus on V2V communication and propose routing solutions for vehicles moving across city streets.

# 3

# A Multihop and Multipath Routing Protocol Using NDN for VANETs

## 3.1 Introduction

In this Chapter, we tackle the problem as described in Section 1.2.1, thus, proposing answers to **RQ1**, which formulates the problem of how to reduce the number of broadcast transmissions of messages in VANETs when the number of interconnected cars is high. The nature of Wi-Fi is always broadcasting messages. This broadcasting leads to waste of resources since unnecessary transmissions occur in the network occupying the channel and reducing the bandwidth. Moreover, broadcasting on the same channel leads to message collisions. Therefore, in this Chapter we investigate how we can reduce the message broadcast transmissions from all nodes to offload traffic from the communication channel.

In particular, we present two Vehicle to Vehicle (V2V) routing protocols using NDN to support vehicular mobility in a dynamic changing networking vehicular environment by reducing broadcast transmissions. In the first routing protocol, named a V2V

Multihop, Multipath and Multichannel routing strategy for VANETs using NDN, **MMM-VNDN**, we filter broadcast messages based on new fields that we introduced into the NDN messages and into the NDN data structures [78]. In the second routing approach, which presents an enhanced work of the first routing protocol and is named **i**mproved **MMM-VNDN iMMM-VNDN** [84], we use both broadcast and unicast transmissions to transmit messages into the network.

The main difference between the two protocols, MMM-VNDN and iMMM-VNDN is the NDN message transmission. In MMM-VNDN we created two new fields in the NDN messages and insert the MAC addresses of the interface of nodes into these fields. Then, we always use broadcast MAC addresses and every node receiving this message broadcasts it into the network. Next, based on these created fields in the NDN messages, we accept or reject incoming messages. Specifically, these newly created fields determine whether a node will accept or reject a message on top of its broadcast transmission.

In iMMM-VNDN, we extract the MAC addresses from the strategy layer of the node, and we use them as fields inside this layer. Thus, we leave the original NDN messages unchanged. The strategy layer of NDN is equivalent to the data link layer of the OSI model. Then, we create unicast transmissions to send messages. In this Chapter, we will introduce both protocols. We will show that by reducing the broadcast transmissions of messages in iMMM-VNDN (since we create unicast transmissions) and by containing no additional information in the messages, iMMM-VNDN achieves better results in terms of content retrieval and latency, compared to MMM-VNDN and other state of the art protocols.

In particular, the contributions of this Chapter can be summarized as follows:

- We develop two Vehicle to Vehicle (V2V) routing protocols, where a node requesting information broadcasts an Interest message to discover potential content sources to create routing entries.
- We exploit the latency and connection timing of these routing entries, and, thus, we develop three different techniques to select a next hop:
  - We distribute the traffic uniformly to all available next hop.
  - We choose next hops based on their latency. To define the latency, we measure how much time passed from the time a node sends an Interest to the time the nodes receives the corresponding Data message.

  – We distribute traffic uniformly to all available next hops with the lowest latency.

We show that the proposed routing protocols allow the network to support the mobility of vehicles, and enable it to adjust to most scenarios that could happen to a VANET, i.e. content source out of range of requester, etc.

The rest of the Chapter is organized as follows: In Section 3.2 we introduce our architecture and present the routing decisions that we developed for both of our protocols. We, then, present the implementation together with our simulation results (Section 3.3). Finally, we draw conclusions in Section 3.4.

## 3.2 Routing

Due to the mobility of nodes in VANETs, we developed two algorithms that discover content sources, create routing entries, and transmit messages based on information that these routing entries provide.

In VANETs, due to the node mobility and the broadcast characteristics of the wireless medium (Wi-Fi), studies (e.g. [71]) propose not to exploit the FIB for any routing decisions. In both of our algorithms, i.e. **M**ultihop, **M**ultipath and **M**ulti-channel for **V**ANETs using **NDN** (**MMM-VNDN**) and ***improved*** MMM-VNDN (**iMMM-VNDN**), the first task is to identify the content source(s) and to create paths between it and the requester node(s), which can be used to receive messages. We define as path a sequence of nodes that a message passes through, starting from the requester node and ending in the destination node. This sequence consists of next hops of the routing entries of nodes that a message passes.

Because of that characteristic, the current NDN logic has been extended and developed as follows: First for MMM-VNDN, fields that contain unique identifiers (MAC addresses) are added in both Interest and Data messages. In particular, for MMM-VNDN we propose two new fields in the Interest and Data message, in addition to the existing NDN header:

- Target MAC Address (TMA) is the destination MAC address of the message. Thus, it shows the next hop that the Interest or Data message will be sent to.

- Origin MAC Address (OMA) is the source MAC address of the message. It shows the network device of the node that the Interest or Data message has been sent from (previous hop).

OMA and TMA assist in identifying intermediate nodes transmitting messages and creating paths, as described Section 3.2.1. In addition, new fields in the PIT and FIB tables of every node have been added to the existing NDN implementation. In iMMM-VNDN we leave the NDN header unchanged, i.e. these new fields are not included in the NDN messages. Instead, we extract the MAC addresses, the origin MAC address and the target MAC address, from the strategy layer at each NDN node. Then, we use the OMA and the TMA as fields inside the strategy layer of each node to perform routing decisions.

### 3.2.1 Routing Decisions

As described in Chapter 2.2, in NDN to route an Interest message, a node checks the FIB table to find available information regarding the next hop.

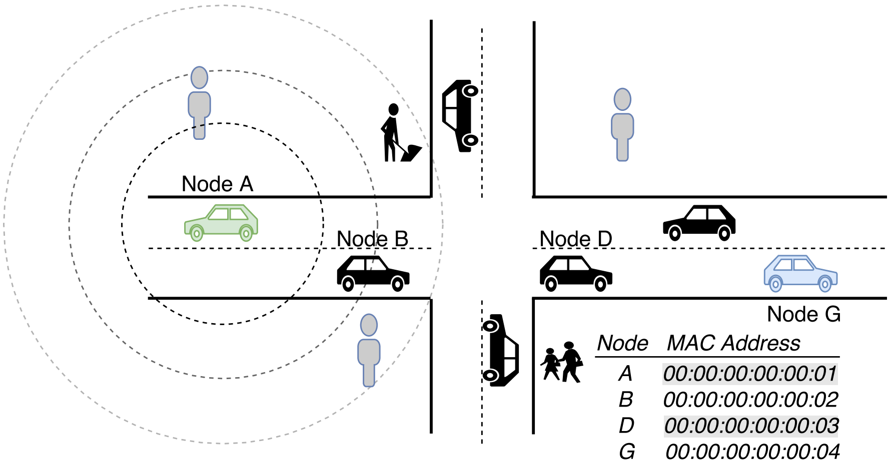
**First Phase - Flooding** First, every node consists of an empty FIB table (there are no entries to the content source). Thus, it should populate its FIB table to find possible routes to the content source.

A node that requests content (requester), broadcasts an Interest message for its request. This Interest message contains the MAC address of the interface of the node that has

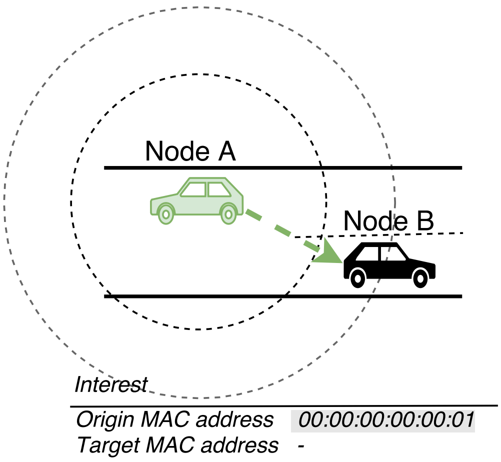
**Algorithm 1** Routing of an Interest from a requester node

```
procedure CHECK FIB
    if FIBEntry = ∅ then
        OriginMAC ← MyMAC
        TargetMAC ← NULL                      ▷ in MMM-VNDN or
        TargetMAC ← BroadcastMAC              ▷ in iMMM-VNDN
        transmit(Interest, nexthop)
    else
        OriginMAC ← MyMAC
        TargetMAC ← Select(nexthop)
        transmit(Interest, nexthop)
    end if
end procedure
```

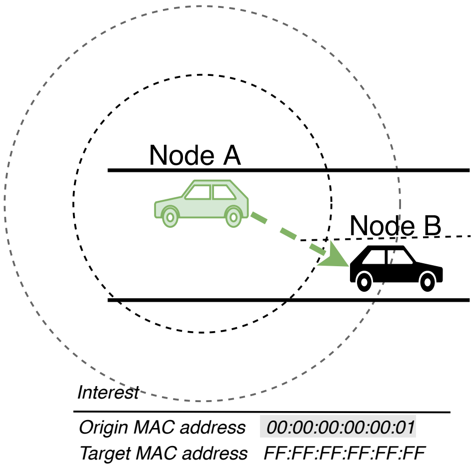
sent the message in the Origin MAC Address (OMA) field (in this case the MAC Address of the requester). In MMM-VNDN the Target MAC Address (TMA) field in this Interest message is empty. In iMMM-VNDN, the TMA in this Interest message is a broadcast MAC address. In Algorithm 1 from lines 2-6, the routing of an Interest from a node with an empty FIB entry is described. The node enters in the OMA its MAC address and in its TMA either null (in MMM-VNDN) or the broadcast MAC address (in iMMM-VNDN).




(a) Broadcasting of the Interest by the requester node for the first time



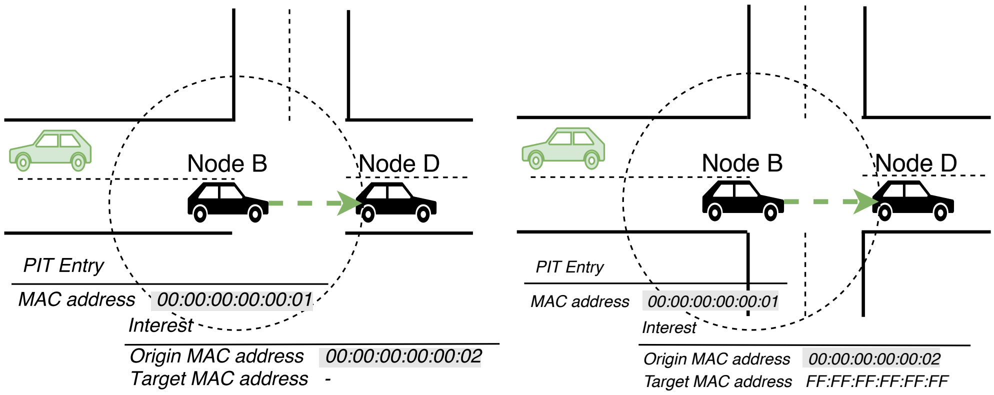

(b) Interest broadcast from node A in MMM-VNDN




(c) Interest broadcast from node A in iMMM-VNDN

Figure 3.1: Flooding phase from requester node A

Let us assume that the topology of a VANET is as shown in Fig. 3.1a. The MAC addresses of the participating nodes are shown. Let us also assume that node A is the requester node, and node G is the content source. First, node A broadcasts an Interest. In MMM-VNDN, the Interest contains two additional fields in its NDN header. In this example, node A enters its MAC address, i.e. 00:00:00:00:00:01 into the OMA field of the Interest message and leaves the TMA field empty, as shown in Fig. 3.1b. In iMMM-VNDN, the Interest contains 00:00:00:00:00:01 as OMA and FF:FF:FF:FF:FF:FF as TMA, which is the MAC broadcast address, as shown in Fig. 3.1c.




(a) Transmission of an Interest from node B in MMM-VNDN

(b) Transmission of an Interest from node B in iMMM-VNDN

Figure 3.2: Differences in processing and transmitting an Interest between MMM-VNDN and iMMM-VNDN

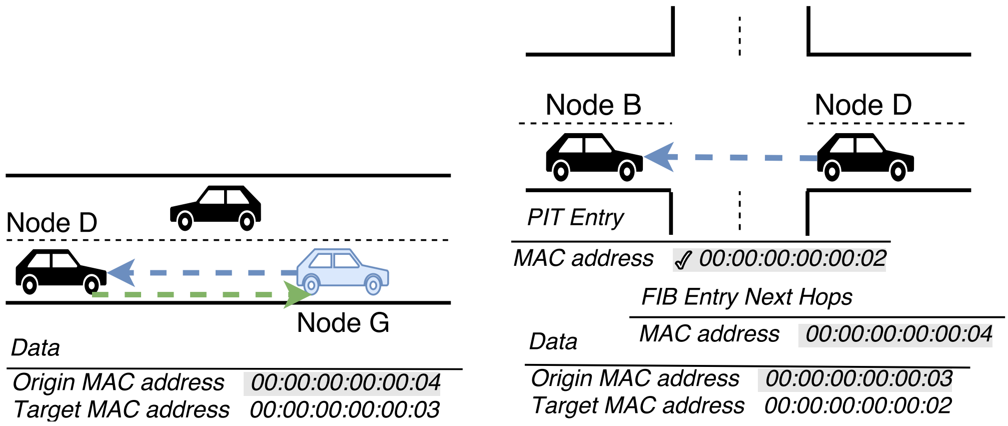
When an intermediate node receives this Interest message, it checks the OMA field, to identify the node that sent the message. Then this node creates a PIT entry, containing this OMA. In MMM-VNDN, the TMA field is empty, thus, this intermediate node continues broadcasting the message without a TMA. As illustrated in Fig. 3.2a, node B that received the broadcast Interest from node A, creates a PIT entry with the OMA field of the received Interest, i.e. 00:00:00:00:00:01. Node B updates the Interest's OMA field to its own, 00:00:00:00:00:02 and because the Interest's Target MAC field is empty, it broadcasts the message. In iMMM-VNDN as seen in Fig. 3.2b, node B, which received the broadcast Interest from node A, creates a PIT entry with MAC address: 00:00:00:00:00:01. Node B updates the Interest's OMA to 00:00:00:00:00:02 and broadcasts the message with TMA: FF:FF:FF:FF:FF:FF.

This process continues for both MMM-VNDN and iMMM-VNDN until this Interest message arrives at the node holding the Data. For both MMM-VNDN and iMMM-VNDN, after receiving the Interest, the content source responds with a Data message and performs the following actions:

(i) It takes the OMA (source address of the previous hop) of the Interest message and inserts it into the Data message as its TMA (destination MAC address).

(ii) It takes its own MAC address (MAC address of the content source) and inserts it into the OMA field of the Data message.

(iii) It sends the Data Message to the network. In MMM-VNDN the content source broadcast this Data Message. In iMMM-VNDN it unicasts the Data message into the network, to all nodes that previously broadcast the Interest message to it.




(a) Node G responding with Data

(b) Node D processes and sends the Data

Figure 3.3: Data processing

The aforementioned process is illustrated in Fig. 3.3a. Assuming that the topology is the same as in Fig. 3.1a, node G, which is the content source, receives an Interest from node D. In MMM-VNDN the Interest has an empty TMA field and the content of the OMA field is 00:00:00:00:00:03. Node G, then, creates a Data message in response to the received Interest message by entering 00:00:00:00:00:03 in the field TMA, and its own MAC address, i.e. 00:00:00:00:00:04, in the field OMA. Then, it broadcasts the Data message. In iMMM-VNDN, the Interest has a broadcast TMA and OMA: 00:00:00:00:00:03. Node G, then, creates a Data message by entering 00:00:00:00:00:03 into the TMA and 00:00:00:00:00:04 into the OMA. Then, it unicasts the Data message into the network.

The process of Data transmission is outlined in Algorithm 2. When an intermediate node receives a Data message it checks if the Data message is meant for itself, i.e. if the TMA (destination address) in the Data message is the same as its own MAC address. If not, the message is discarded. If the TMA of the message and the node's MAC address is the same, the node checks the PIT. If there is no matching PIT entry, the Data message is discarded. If there exists a PIT entry, the node performs the following actions:

(i) The node creates (or updates), a new FIB entry that contains the OMA of the Data message (source address of the previous hop of the message).

(ii) The node updates the OMA of the Data message to contain its own MAC address.

(iii) The node checks the PIT entry that has been created from the respective Interest and sets the TMA of the Data message to the MAC address of the PIT entry.

(iv) The node enters the Data packet in its CS.

(v) The node sends the updated Data message to the network. In MMM-VNDN this transmission is broadcast. In iMMM-VNDN the transmission is unicast.

**Algorithm 2** Routing of a Data message from an intermediate node

```
procedure RECEPTION OF DATA
    if PITEntry ≠ ∅ then
        if TargetMAC ≠ MyMAC then
            Create/Update(FIBEntry, OriginMac)
            return
        else
            Create/Update(FIBEntry, OriginMac)
            OriginMAC ← MyMAC
            TargetMAC ← PITEntry(nexthop)
            DeletePITEntry
            transmit(Data, nexthop)
        end if
    end if
end procedure
```

In Algorithm 2, we prevent loops and limit the transmission of the Data message. In MMM-VNDN and iMMM-VNDN, we check if a node has a PIT entry with the same name as the Data message. If the node does not have a PIT entry, then the Data message is discarded. In addition, in both algorithms, we delete the PIT entry before the Data routing. By this, loops are avoided: If the Data message is received again in the same node, there will be no PIT entry, and the Data message will be discarded.

To continue with our example, the Data message node G sent in Fig. 3.3a is now incoming in node D. As shown in Fig. 3.3b node D first checks if a PIT entry exists for this Data message. Since node D transmitted the corresponding Interest (as shown in Fig. 3.2a), a PIT entry exists. Node D then starts processing the Data, by checking the TMA field, to see if this Data message is meant for it. The TMA field of the Data message is set to 00:00:00:00:00:03 (as shown in Fig. 3.3a) and since it is the same as nodes D's MAC address, node D accepts the message and processes it. After, node D creates or updates a FIB entry with the OMA field of the Data, i.e. 00:00:00:00:00:04. Finally, node D updates these two fields of the Data message: It sets as TMA the address that exists in the PIT, i.e. 00:00:00:00:00:02, and as OMA its own MAC address, i.e. 00:00:00:00:00:03.

This process of broadcasting the Interest in every node continues until the requester receives the first Data message. Since multiple nodes transmit the same Interest, the responding Data message will arrive at the requester node many times through different nodes. When the requester receives the Data message, it creates or updates a FIB entry containing the OMA field of the Data message, i.e. the MAC address of the previous node that transmitted the message. Hence, the FIB was extended with two additional fields. Fig. 3.4 shows the first path that has been established from the requester, node A, to the content source, node G. Node B, node C, and node D are intermediate nodes that assist in the message transfer since there is no direct connection between node A and node G.

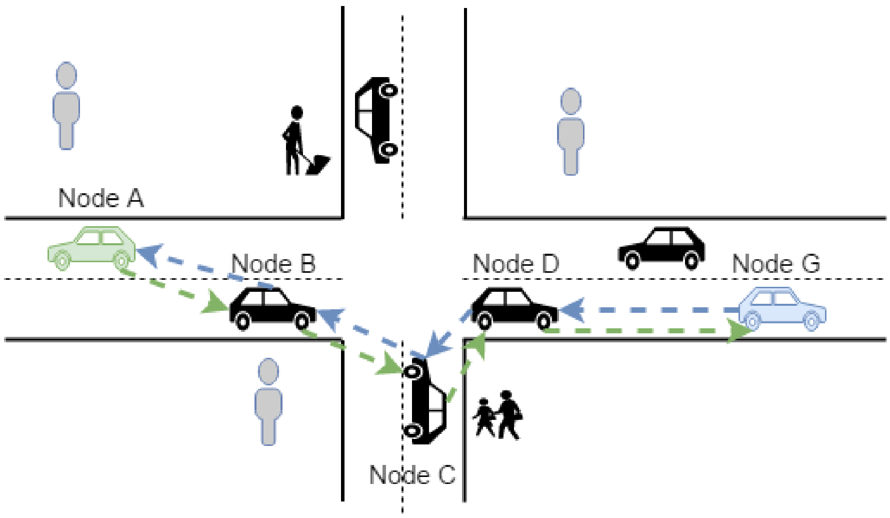
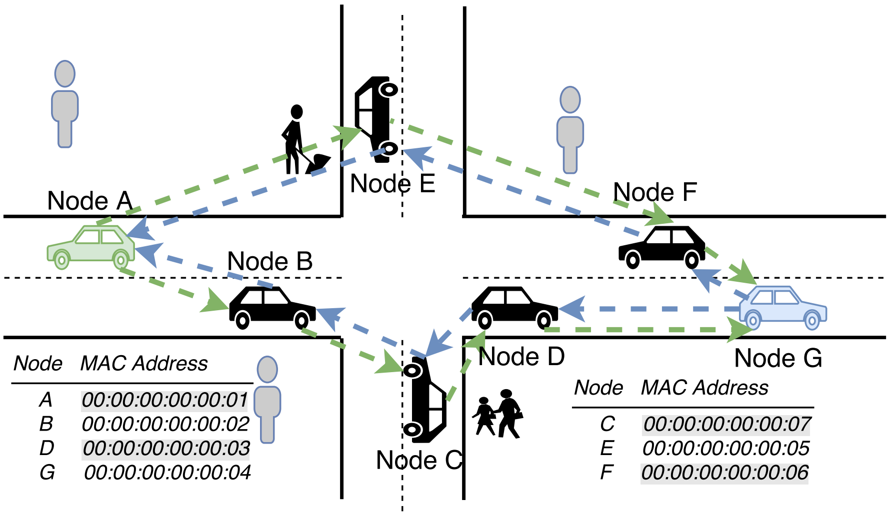
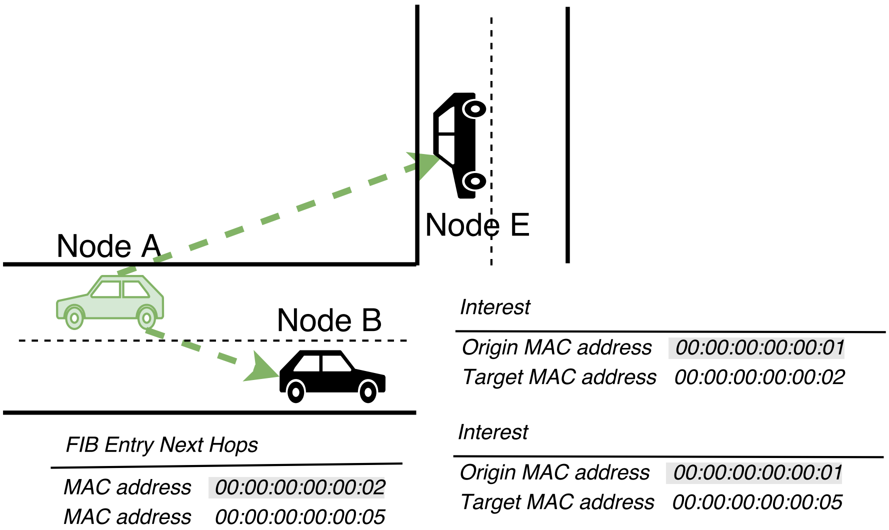



Figure 3.4: Established connection through PIT and FIB entries

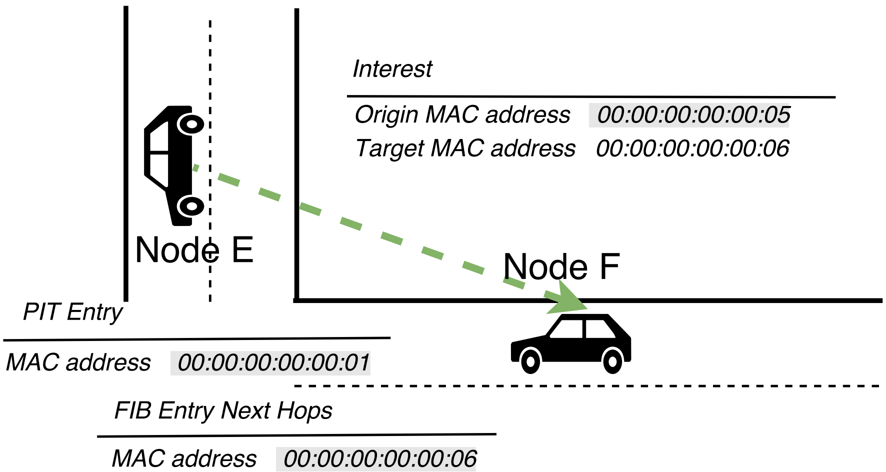
**Second Phase - Routing Based on FIB** Subsequently, when the requester node sends the second Interest, it checks the FIB table to identify possible routes. It will search the FIB for MAC addresses of next hops, i.e. MAC addresses of next nodes to transmit the Interest. The requester then selects a next hop and sets the Interest fields as follows: The OMA field containing the current node's MAC address, and the TMA field containing the MAC address of the next hop that has been selected.




(a) Paths that are established after a flooded Interest




(b) Interest transmission from node A




(c) Interest processing and transmission from node E



Figure 3.5: Routing based on FIB entries

Fig. 3.5a shows the VANET topology and the paths that are established after an Interest was flooded in the network by broadcasting the Interest from every node. Node A will continue its content retrieval by sending a second Interest into the network.

First, as shown in Fig. 3.5b, node A checks its FIB, where it identifies two possible next hops: 00:00:00:00:00:02 and 00:00:00:00:00:05. It selects one (the selection criteria is described in the Section 3.2.2), and updates the TMA field of the Interest, to the selected next hop MAC address, e.g. 00:00:00:00:00:05. The OMA field is set to its own MAC address, 00:00:00:00:00:01. It then sends (broadcast with MMM-VNDN and unicast with iMMM-VNDN) the Interest into the network.

When a node receives an Interest containing a non-empty or non-broadcast TMA, it will check if the latter is the same MAC address as its own. If the Target MAC is different, it will discard the message. In case the MAC address is the same as its own, the Interest is meant for this node, and the node will accept it for further processing. First, if there is no match in the node's CS, the node will enter the OMA from the Interest message to the PIT. Then, it will check its FIB table in order to identify possible next hops. Next, it will send the message with the updated OMA field that is set to its own MAC address, and a TMA field that is set to the MAC address of the chosen next hop.

Fig. 3.5c shows this process, where an incoming Interest from node A has arrived at node E. Node E will check if the Interest is meant for it, i.e. if the TMA of the Interest is the same as its own, i.e. 00:00:00:00:00:05. Since these MAC addresses are the same, node E will create a PIT entry with the Interest's OMA, and then search its FIB for possible next hops. Since there is only one next hop, with MAC address 00:00:00:00:00:06, node E selects this MAC address from the FIB, and inserts it into the TMA of the Interest message. It then updates the OMA field of the Interest to its own MAC address, 00:00:00:00:00:05, and sends the message into the network. This process continues until the Interest message arrives at the content source. The responding Data message follows the reverse path, as described in the flooding phase in Section 3.2.1 Algorithm 2. The Data will be transmitted by nodes if the nodes contain a PIT entry with the same prefix as the Data message. In this case, the nodes will enter into the OMA field their MAC address, and into the TMA field, the MAC address extracted from the PIT entry.

### 3.2.2 Next Hop Selection

When the flooding, using broadcast for Interest transmissions occurs, the Interest message passes through many intermediate nodes and subsequently, the Data message will also pass through some of these nodes. The requester(s) and intermediate nodes will receive the same Data message by different nodes and will create FIB entries that will contain multiple MAC addresses. When a node wants to route an Interest message, it checks the FIB to find next hops in its routing entry, i.e. MAC addresses of nodes that have potential a connection to the content source. If many next hops exist, a node should select one or more of them. To perform so, the FIB will have additional fields to the existing FIB data structure, as shown in Table 3.1. For the selection of the next hop for the Interest, we propose three approaches.

Table 3.1: FIB Entry next hop additional fields

| MAC address | Latency(ms) | Counter |
|---|---|---|
| 00:00:00:00:00:02 | 100 | 0 |
| 00:00:00:00:00:05 | 50 | 0 |

(i) When a FIB next hop is created, the counter field is set to 0. Every time this next hop is chosen for transmitting an Interest, the counter field is incremented by one. The first approach, named Uniformly selected PATH (**UPath**), is selecting the next hop field that corresponds to the lowest counter. This ensures that we will distribute the traffic to all possible next hops. When there are multiple next hops with the same counter, the selection is based on the last added next hop of the FIB table. For instance, in the example shown in Fig. 3.5a, there are two paths. Since both next hops have the same number in their counter field (Table 3.1), node A will choose 00:00:00:00:00:05, because it was the last that was added into the FIB.

(ii) The second approach, named Smallest Latency PATH (**SLPath**), is to choose the next hop that has the lowest latency. Latency is defined as the time that has passed from the transmission of the Interest message to the reception of the Data message in a specific node. Then this time is included in the respective FIB entry. In this case, based on Table 3.1, node A will choose the second next hop, because of the lowest latency, with a MAC address of 00:00:00:00:00:05.

(iii) The third approach is based on the combination of (i) and (ii) and is named Uniform Selection of Lowest latencies PATH (**USLPath**). A counter in each next hop is added. Each time when a next hop is selected (selection of the next MAC address), this counter is increased by one. When multiple next hops exist, which have been added almost at the same time and have not been selected for transmitting an Interest, their counters will have the same value. In that case, the next hop with the lowest latency is chosen. For the values that are shown in Table 3.1, we assume that their counter field is zero. Hence, the Latency field will be checked, and the next hop with the lowest latency will be selected, i.e. 00:00:00:00:00:05. Loop problems are avoided since the selection of next hops is based on a fair distribution. For instance, in Table 3.1 the node sends the Interest to 00:00:00:00:00:05. Then the node increases the counter field on Table 3.1 of the next hop 00:00:00:00:00:05 by 1. We assume that the Interest is coming back to this node. This time the node checks the FIB and sends the Interest to 00:00:00:00:00:02, since this is the next hop with the lowest value in the counter field.

The next hops in the FIB entries that result from overheard Data are part of this process. Such next hops are chosen only if there does not exist another next hop, or if other next hops have been selected many times, to avoid collisions. As shown in Table 3.1, the counter for both the next hops is zero. If the first entry (00:00:00:00:00:02) were a result of an overheard Data, this MAC address (00:00:00:00:00:02) would be added to the FIB entry with a higher counter. This distinction allows prioritizing these next hops from others and select them only when they are the only ones available.

### 3.2.3 Creation of Routing Entries

Since the topology in VANETs is constantly changing, created paths from a requester to a content source may break unexpectedly. To discover new routes and new content sources, the chosen approach is to flood an Interest every few seconds. In MMM-VNDN the Interest TMA will be empty and in iMMM-VNDN the Interest will contain in its TMA the broadcast MAC address. By this periodic flooding, the FIB is populated regularly with new and active connections. When new vehicles in the network are discovered, they are also used for transmission of messages, and new routes that include them are created. Every time a node receives an Interest an empty or a broadcast MAC address, the node deletes its FIB table entries. In this way, we control the FIB size and update it accordingly every time a new route is created.

## 3.3 Performance Evaluation

### 3.3.1 Simulation Environment

#### Simulation Scenarios

To evaluate the routing protocols described in Section 3.2, we used the three different next hop selection options, which meaning and description is presented in Section 3.2.2:

(i) **UPath**:The first set of results is performed for selecting next hops from the FIB table that have the lowest counter value. When multiple next hops have the same counter value, the most recently added is selected.

(ii) **SLPath**: The second approach is based on the average latency of the next hops in the routing entries. The next hop with the lowest latency is chosen for transmitting an Interest.

(iii) **USLPath**: The last approach is based on the combination of the previous two. As mentioned in Section 3.2.2, we select the next hop with the lowest counter. If there are multiple next hops with the same counter, the one with the lowest latency is selected.

#### Simulation Parameters

Moreover, to evaluate our routing protocols we used the ndnSIM simulator [18], v2.0. NdnSIM is a software module providing the basic NDN implementation [18, 19, 106] for the ns-3 network simulator [8]. To obtain the network traffic simulation we used SUMO [36]. For the topology, for MMM-VNDN we chose the MANHATTAN scenario, 1km x 1km, with the number of nodes (cars) varying from 60 to 100. Each vehicle is equipped with three interfaces, to send and receive Interest and Data messages simultaneously, using Wi-Fi 802.11a (We use IEEE 802.11a since 802.11p is not available in ndnSIM v2.0). For iMMM-VNDN we have chosen two different topologies to evaluate the algorithms, the Manhattan map, and the Luxembourg map [47]. In the Manhattan map, the number of nodes (cars) chosen is varying from

Table 3.2: Simulation Parameters

| STRATEGY | PROPAGATION LOSS MODEL | STANDARD-TRANSMISSION RANGE | DATA RATE | CHANNEL BANDWIDTH |
|---|---|---|---|---|
| MMM-VNDN | Three Log Distance and Nakagami | IEEE802.11a 200m | 24 Mbps | 20MHz |
| iMMM-VNDN | Two Ray Ground | IEEE802.11p 250m | 6 Mbps | 10MHz |
| Broadcasting | Three Log Distance and Nakagami | IEEE802.11a 200m | 24 Mbps | 20MHz |
| Best-route | Three Log Distance and Nakagami | IEEE802.11a 200m | 24 Mbps | 20MHz |
| NCC | Three Log Distance and Nakagami | IEEE802.11a 200m | 24 Mbps | 20MHz |
| CCVN | Two Ray Ground | IEEE802.11p 250m | 6 Mbps | 10MHz |
| CODIE | Two Ray Ground | IEEE802.11p 250m | 6 Mbps | 10MHz |

20 to 100 and the average speed is around 15m/s. In the Luxembourg map, we have chosen an area of 1km x 1km in the city centre. Luxembourg traces are available for 24 hours. We have chosen different times during these 24 hours to extract the mobility traces. Then, in these different mobility traces, we have extracted the density of vehicles, varying from 109 to 396. The average speed of cars depends on the time slot when the mobility traces were extracted. The parameters of the algorithms are shown in Table 3.2. In Table 3.2 the standard defines the transmission range, the data rate and the channel bandwidth. We used the IEEE 802.11a standard for MMM-VNDN, Broadcasting and NCC strategies because these strategies were developed in ndnSIM v2.0 and ndnSIM v2.0 is not compatible with IEEE 802.11p. The propagation models include the Three Log Distance model, which is a log distance path loss propagation model with three distance fields [4], the Nakagami model, which accounts for the variations in signal strength due to multipath fading [5], and the Two Ray Ground model, where the gain of the signal depends on the reflection of the signal from

roads [6].

For the next hop selection options chosen (Section 3.3.1(i), (ii), (iii)), one node is sending Interest messages and there exists one content source in the network. For each of the above next hop selection options, we first compare MMM-VNDN with the flooding strategy, which next hop an Internet message into the network. Then, we compared iMMM-VNDN with MMM-VNDN to show the advantages that the improved protocol offers. Finally, for the third next hop selection option, we compared iMMM-VNDN with the broadcasting strategy, which broadcasts every Internet message into the network, the best route strategy, which chooses the best face to send an Interest according to its cost [19], the NCC strategy [7], which chooses the best face to send the Interest and tracks the RTT of faces, the Content-Centric Vehicular Networking (CCVN) algorithm [24], described in Section 2.3 and the Controlled Data and Interest packet propagation strategy (CODIE) [21], where the Interest contains a hop counter and the Data propagation is controlled by this hop counter. The presented results have a confidence interval of 95%. Since the proposed algorithms are developed for VANETs, new nodes could enter the network at any time. This leads to intermittent connections and consequently to breaking the paths from the requester to the content source. To cope with these problems, we broadcast *one* Interest message every 10 seconds. By broadcasting only one instead of every Interest, routing entries are created or updated and broadcast storms are avoided. This is accomplished by not broadcasting the network with unnecessary Interest transmissions.

For all scenarios chosen, we experiment with three different Interest Lifetime (IL) times, i.e. the time that the Interest is transmitted into the network before it expires. We note that the IL lifetime depends on the application, i.e. how much an application should wait before requesting again its Data. We propose our algorithm in scenarios where vehicles request infotainment data, e.g. a video of a road, where there is a tolerance for a small delay in receiving the Data. Therefore, we choose different ILs: 4 seconds, 8 seconds and 12 seconds. We consider these values to be realistic for infotainment application with delay constrains. Moreover, such small values are widely used in the literature [29, 32]. We highlight here that using large values for the I.L. could result in prevention of forwarding similar Interests because the request is already pending (a node has an entry to its PIT.

### Simulation Metrics

To evaluate the routing protocols three metrics were used:

- *Interest Satisfaction Rate (ISR)* describes the number of received Data messages that were received by the requester divided by the total number of Interest messages being sent.
- *Average Latency*: The average latency for all received Data messages describes the average time that passed from the time a requester sent an Interest message to the time that the requester received the Data message. When a Data message has not been received until its expiration time, the requester node retransmits the Interest message and the initial time is reset.
- *Average Jitter*: Jitter is defined as the mean deviation of the difference in packet spacing at the receiver node compared to the sender for a pair of packets [64].

The above metrics describe the main characteristics that we consider important to the V2V communication in a VANET for an infotainment application. ISR is a core metric, but it is insufficient, when a VANET application has delay constraints. Hence, latency and jitter are being measured in our experiments.

## 3.3.2 Simulation Results

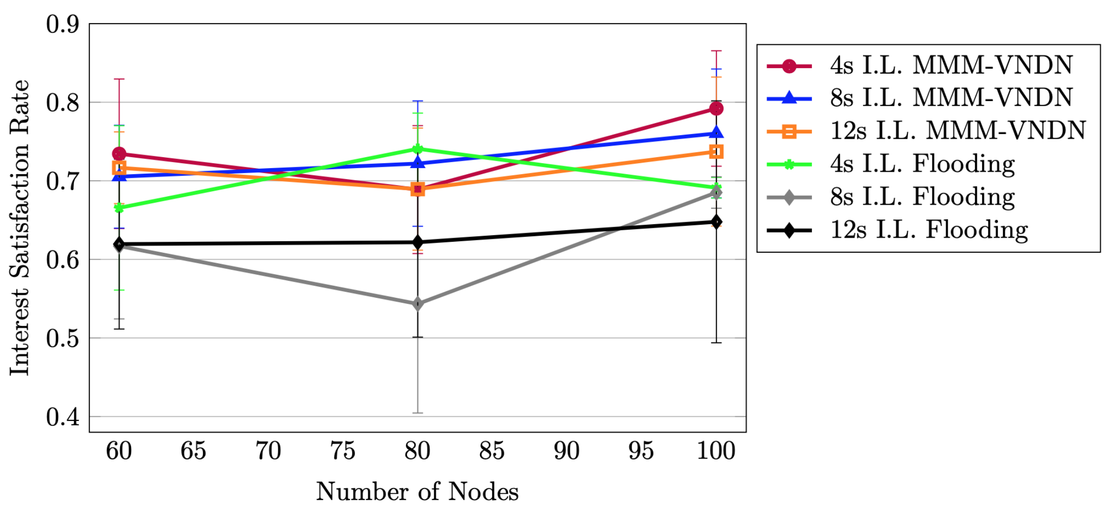



Figure 3.6: Interest Satisfaction Rate for the UPath

The first set of results that show how MMM-VNDN performs are presented in Figs. 3.6–3.11. For UPath, the results of the ISR are shown in Fig. 3.6. The proposed next hop selection option achieves up to 10% higher delivery rate, with lower latency, compared to the broadcasting approach. The average latency is presented in Fig. 3.7 and fluctuates from 2 to 5 seconds, compared to the broadcasting routing protocol, where it is around 14 seconds. These results indicate that by selecting next hops, the network resources are released and the network is less congested.

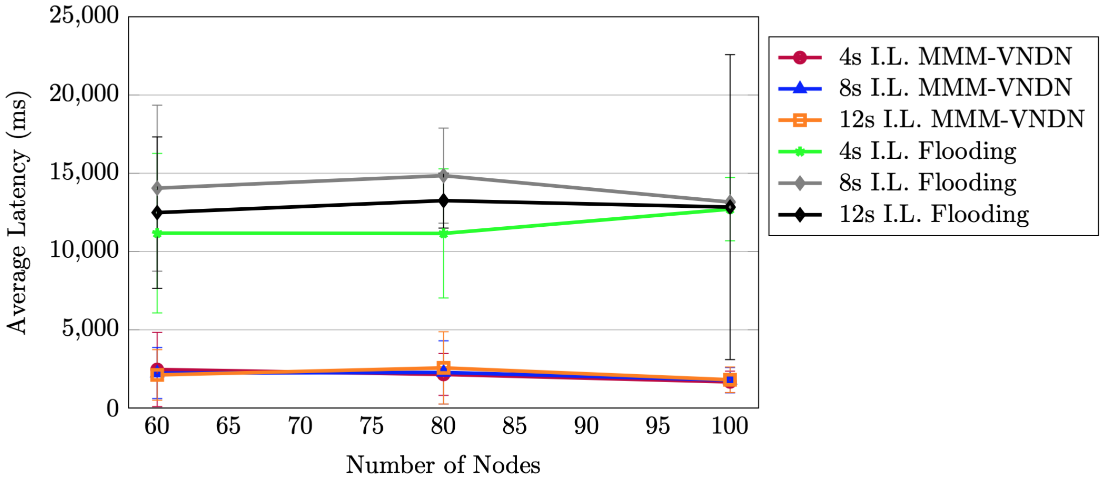



Figure 3.7: Average Latency for the UPath

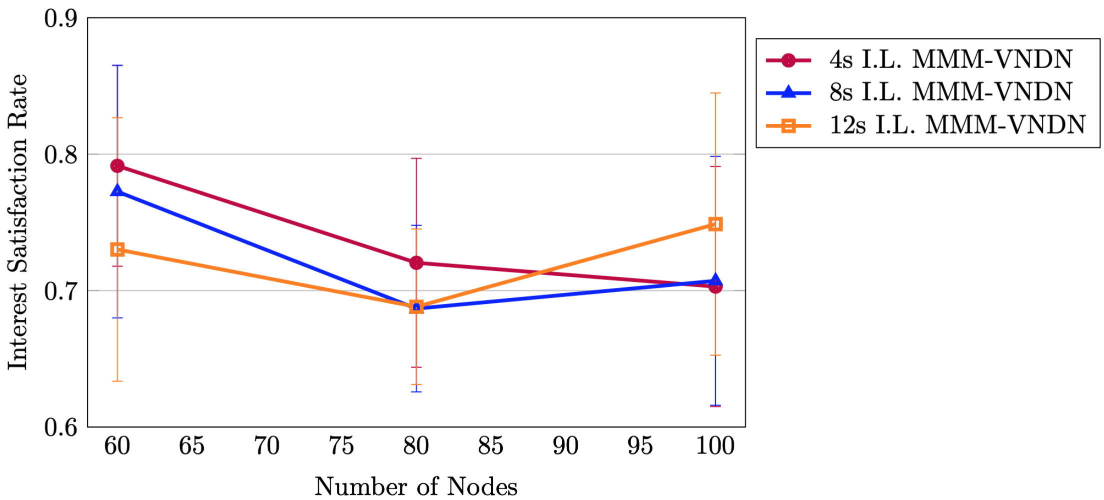



Figure 3.8: Interest Satisfaction Rate for the SLPath

In SLPath, we see that the ISR is better when there is a short Interest Lifetime of 4 seconds, and the node density is low. The performance is also increased, when the Interest Lifetime and the node density of the network are higher (Fig. 3.8). This is due

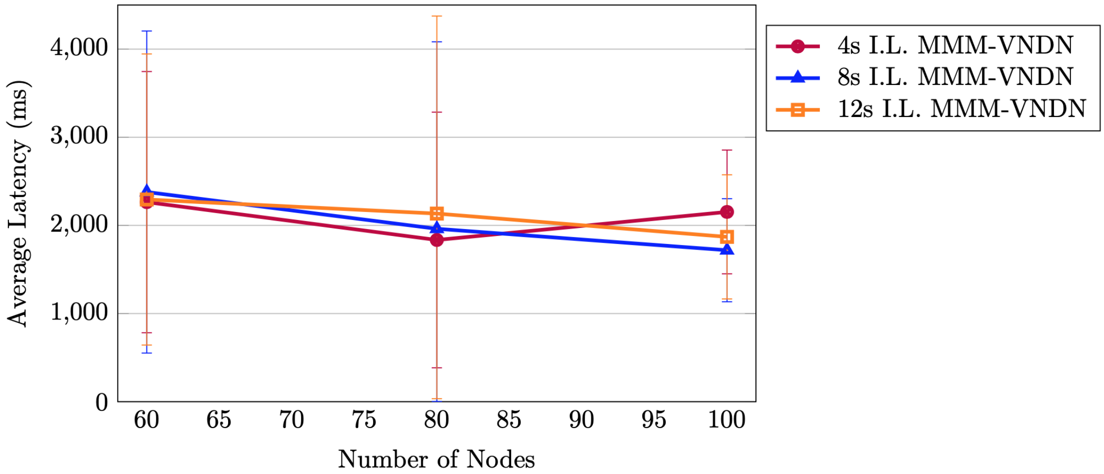



Figure 3.9: Average Latency for the SLPath

to the fact that in a sparse VANET when the Interest expires, the retransmissions are fewer because the number of nodes is also small. When the VANET is dense and the Interest expires, the number of nodes that will route the Interest is high, leading to lower ISR. In the other case, when the Interest Lifetime is at 12 seconds, the Interest is more likely to reach the content source before it expires. The average delay is kept at the same level, around two seconds, for both of these cases (Fig. 3.9).

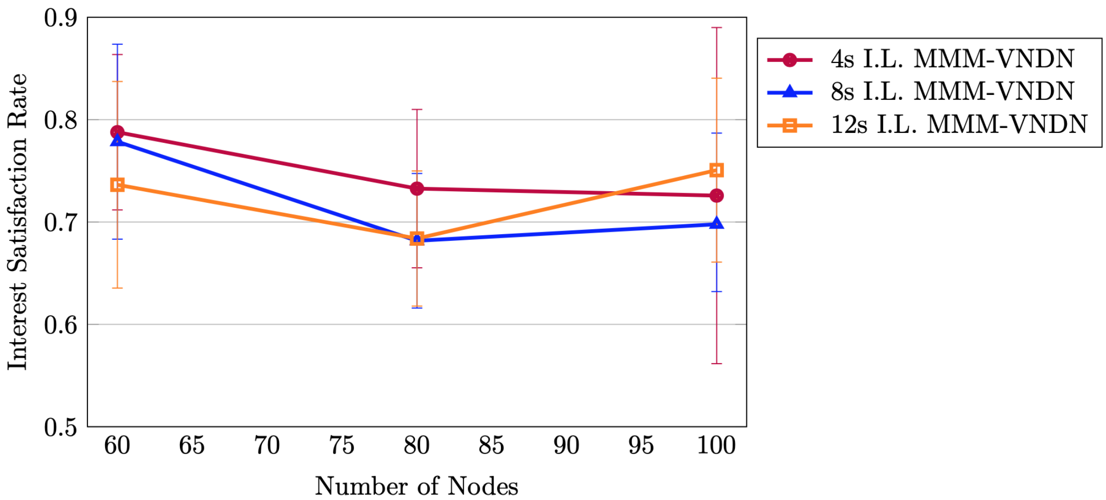



Figure 3.10: Interest Satisfaction Rate for the USLPath

In the USLPath, we achieve the best results compared to the other two. This is due to the fact, that the selection of next hop is based on both the average latency that this hop provides, with the latest time where the connection with this hop is established. Fig. 3.11 shows that the latency for all Interest Lifetime values is kept at an almost

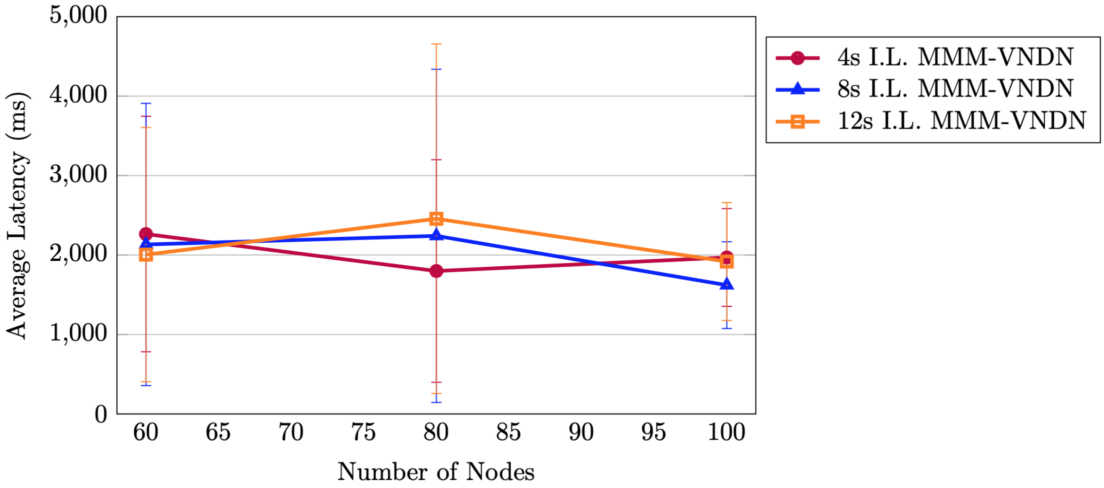



Figure 3.11: Average Latency for the USLPath

steady level of 2 seconds, independent from the density of the network. In addition, we notice that again the performance of our network is higher than a broadcasting approach (that is illustrated in Fig. 3.6 and in Fig. 3.7), both in terms of the ISR (Fig. 3.10) and the Average Latency (Fig. 3.11). The overhead of the network that the broadcasting approach produces is high, and thus the network is congested, which leads to longer waiting time and lower ISR.

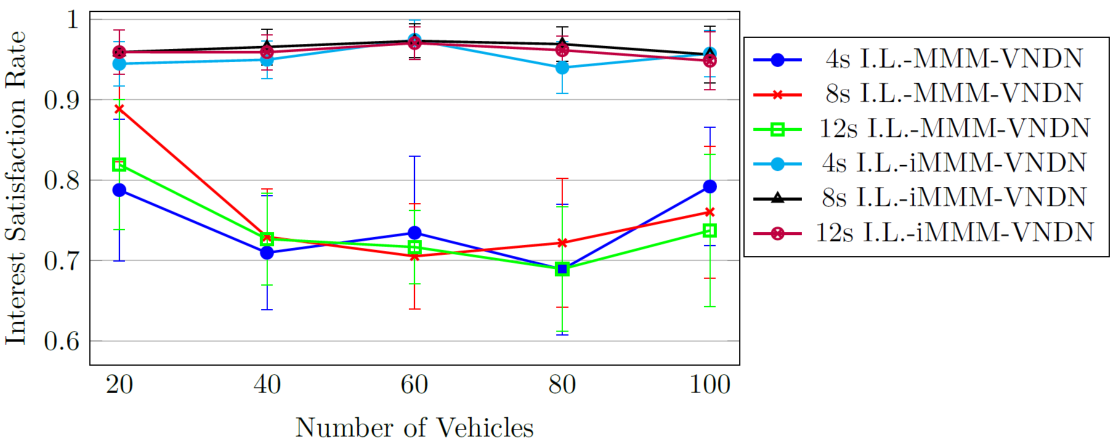



Figure 3.12: ISR in Manhattan map for the UPath

For the second set of our results, first, we compare the two routing protocols for the first next hop selection approach (UPath) in Figs. 3.12–3.14. iMMM-VNDN keeps the ISR above 93% compared to MMM-VNDN that keeps the ISR above 70%, as shown in Fig. 3.12. This is due to the fact that the decision of processing or discarding a message is performed in the strategy layer of the NDN stack, thus allowing more control over the incoming messages. In addition, the average latency is lower for iMMM-VNDN than for MMM-VNDN and remains in between 2.5 - 3 ms, regardless of the number of nodes Fig. 3.13. iMMM-VNDN performs worse in terms of jitter, which stays almost at 0.5 ms independent of the density of the nodes Fig. 3.14.

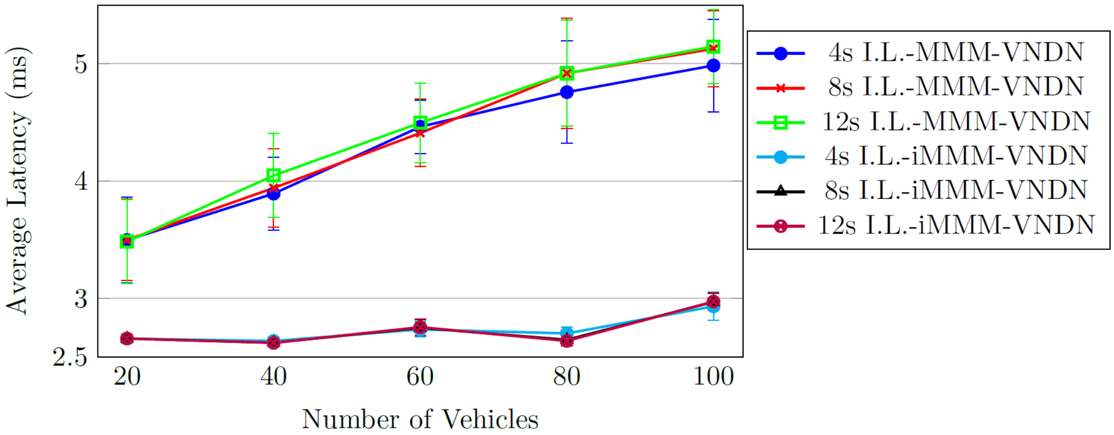



Figure 3.13: Average Latency in Manhattan map for the UPath

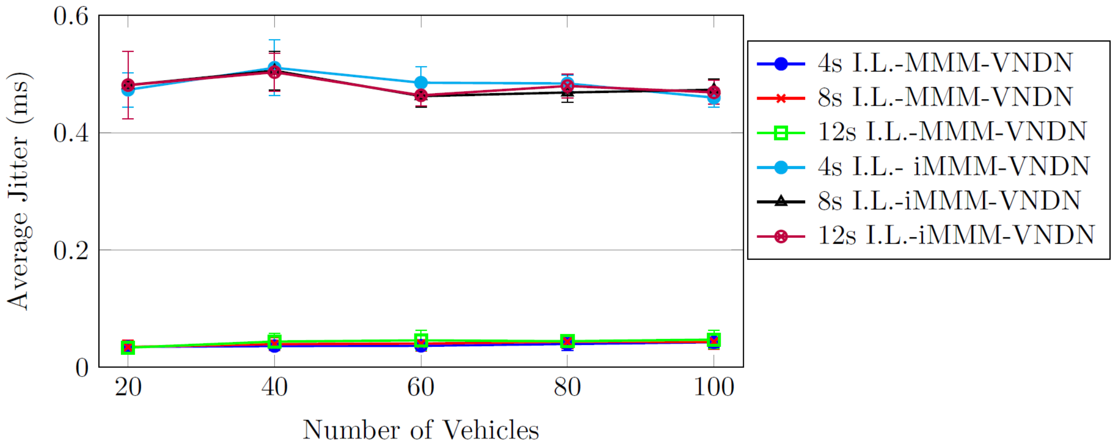



Figure 3.14: Average Jitter in Manhattan map for the UPath

In the second presented next hop selection approach (SLPath) in Figs. 3.15–3.17, we observe that for both protocols the ISR fluctuates. Generally, iMMM-VNDN performs better than MMM-VNDN, but it is clear that choosing the next hop with the lowest latency does not guarantee that the path will be valid. In contrast, because of the path breaks, we see that the ISR for both approaches is lower than in Fig. 3.12. For all algorithms, we also observe a drop in the ISR, when 40 vehicles exist in the network. For iMMM-VNDN in particular, since the protocol is based on unicast transmissions when paths break, nodes retransmit the Interests through invalid paths, since the next

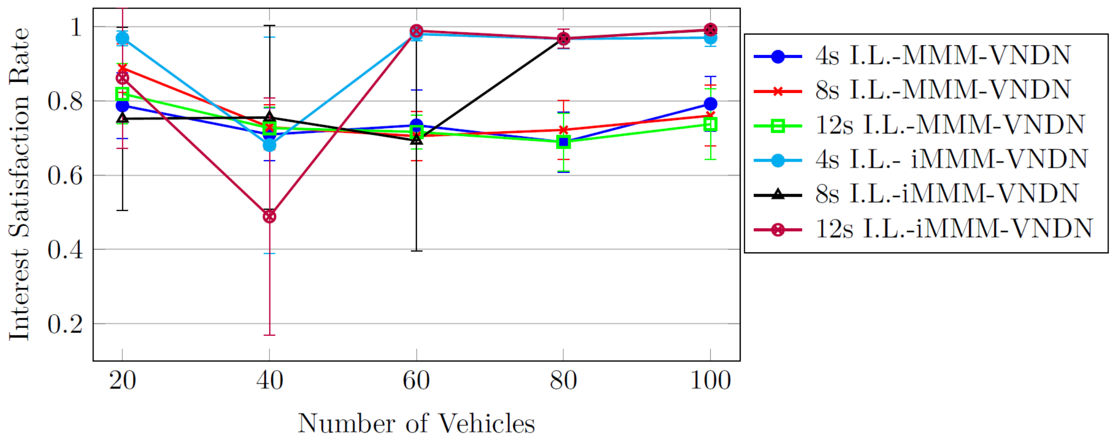



Figure 3.15: ISR in Manhattan map for the SLPath

hop selection is based on the latency field in the FIB entry chosen. Hence, unicast transmissions fail because nodes (since the number of vehicles is sparse) may have fewer FIB entries and most of them are invalid. Also, despite the fact that the goal was to decrease the average latency, the average latency is kept at the same levels as in Fig. 3.13. The average jitter also remains the same for both algorithms, and we observe that there are small fluctuations in jitter for low ISR.

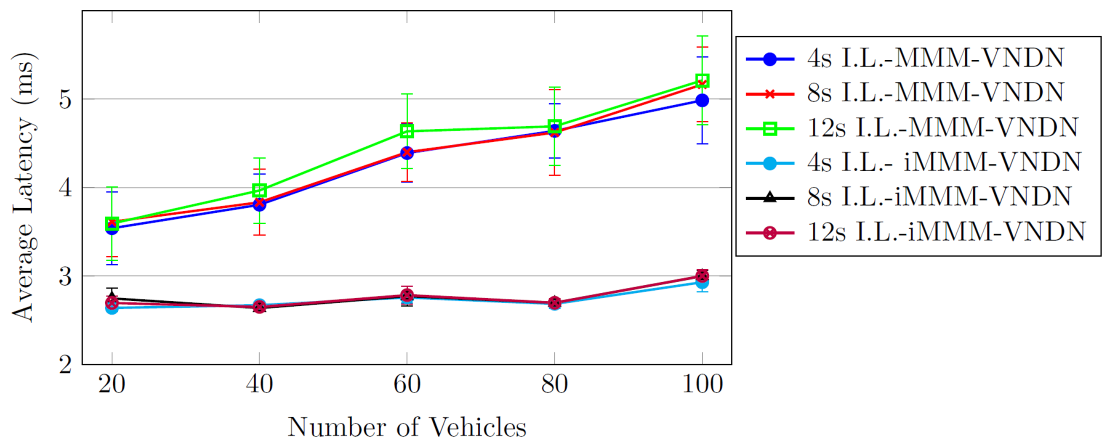



Figure 3.16: Average Latency in Manhattan map for the SLPath

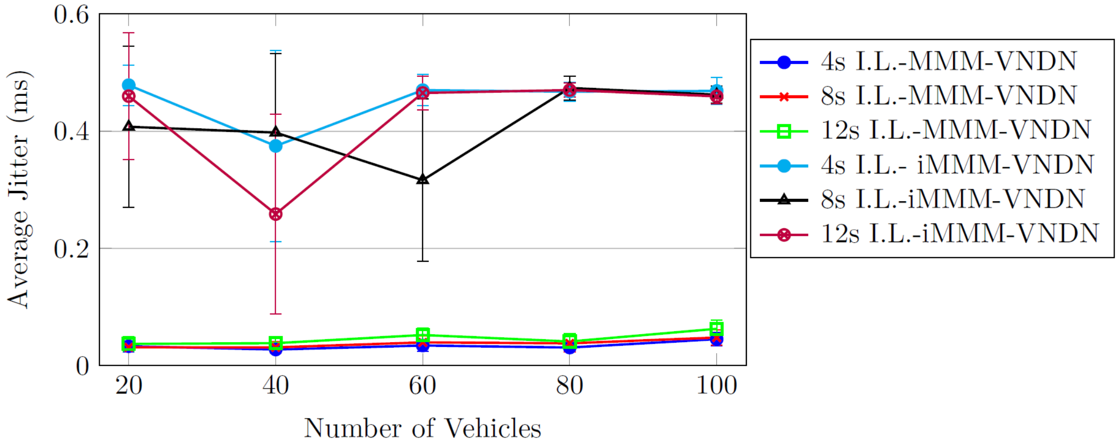



Figure 3.17: Average Jitter in Manhattan map for the SLPath

For the third next hop selection approach (USLPath), we achieve the best results compared to the other two. This is because the selection of a next hop is based on the average latency of the next hop of the routing entry together with the latest time when the entry was established. The results for the ISR are shown in Figs. 3.18–3.20. iMMM-VNDN achieves the highest ISR compared to other protocols, independent from the I.L. value. We observe in Fig. 3.18 a drop in the flooding and the NCC strategy. This is because the network is sparse and even through always broadcasting, vehicular paths break and Data messages cannot be delivered back to the requester node. We also highlight that the ISR of MMM-VNDN strategy fluctuates from 70-80% and is almost 10% higher compared to broadcasting. In Figs. 3.18–3.20, we observe that

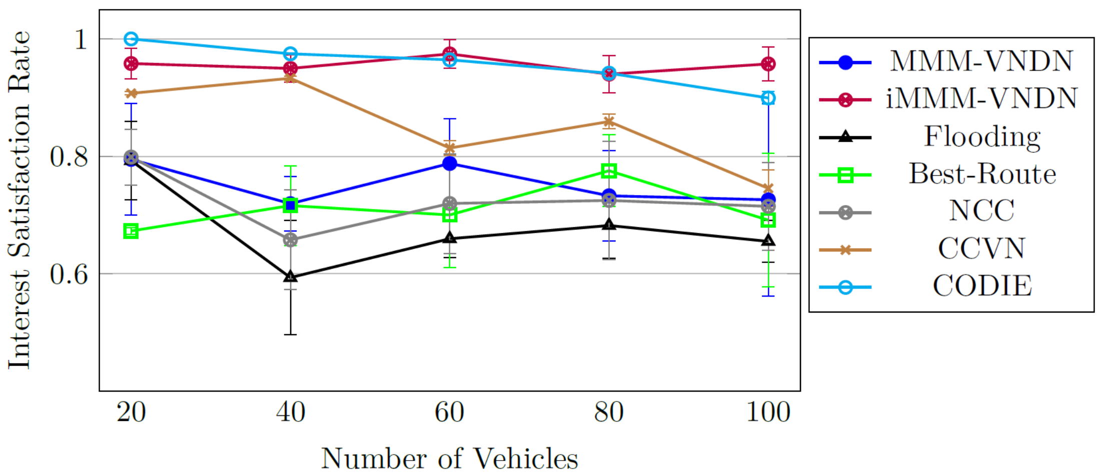



Figure 3.18: ISR in Manhattan map for the USLPath for Interest Lifetime of 4 seconds

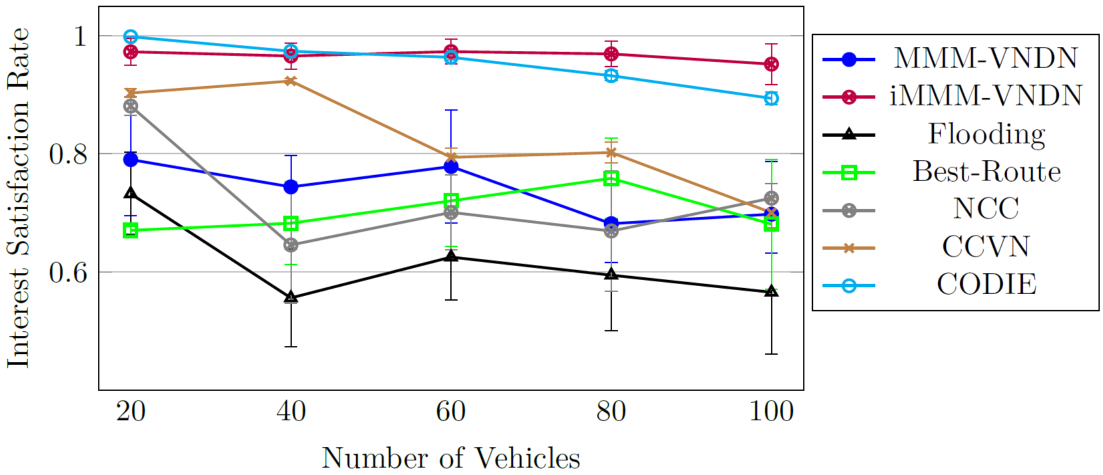



Figure 3.19: ISR in Manhattan map for the USLPath for Interest Lifetime of 8 seconds

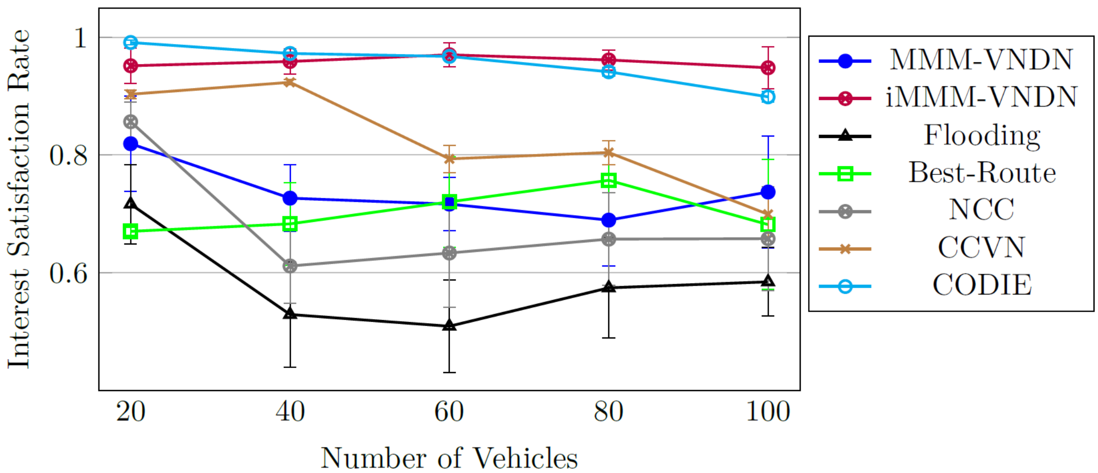



Figure 3.20: ISR in Manhattan map for the USLPath for Interest Lifetime of 12 seconds

best-route and CCVN strategies have higher ISR in some cases than MMM-VNDN. But, as shown in Figs. 3.21–3.23, CCVN's delay of the delivered content object is up to 20 times higher.

Figs. 3.21–3.23 show the average latency of each strategy. The results indicate that our algorithms together with the best-route strategy have the lowest latency for all network sizes. The average latency fluctuates for MMM-VNDN from 3 to 5 ms and for iMMM-VNDN from 2.5 to 3 ms. This difference comes from the number of messages that are being delivered. Higher ISR for iMMM-VNDN means that more messages exist in the network. Thus, the possibility of collisions is higher. In contrast, other strategies achieve much higher delay than both of our proposed protocols. By considering the

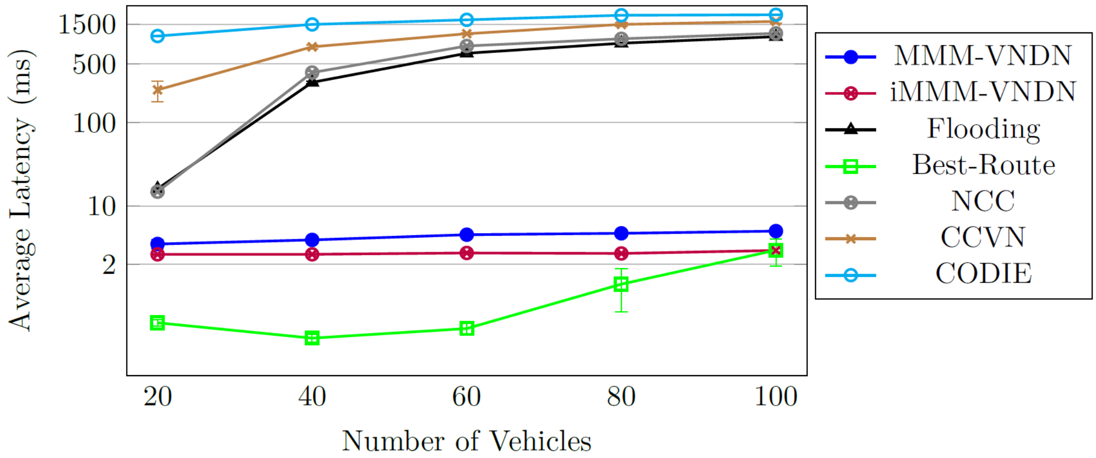



Figure 3.21: Average Latency in Manhattan map for the USLPath for Interest Lifetime of 4 seconds

results of the ISR graph in Figs. 3.18–3.20 we manage to deliver more requested content objects. We reduced the latency by selecting appropriate next hops. Furthermore, the network resources are released, because not all nodes participate in message transmissions, and the network is less congested.

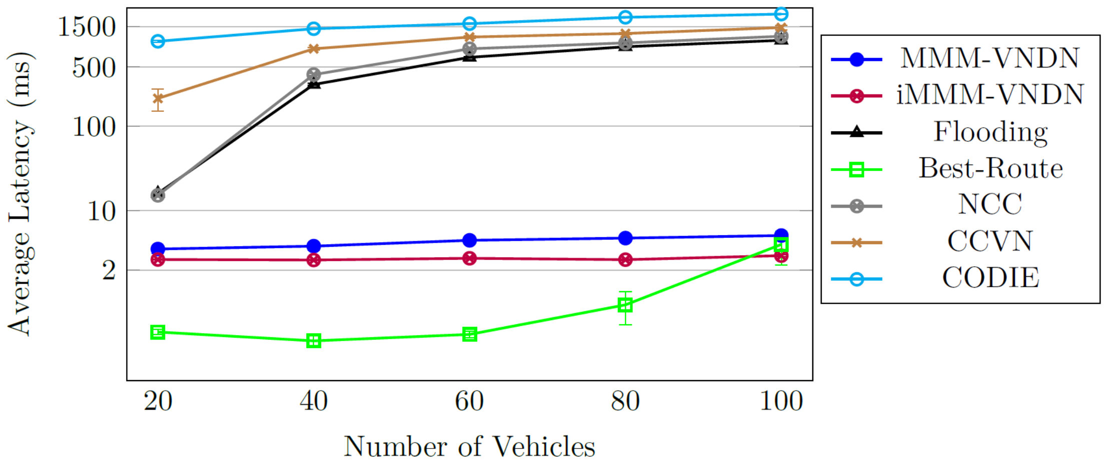



Figure 3.22: Average Latency in Manhattan map for the USLPath for Interest Lifetime of 8 seconds

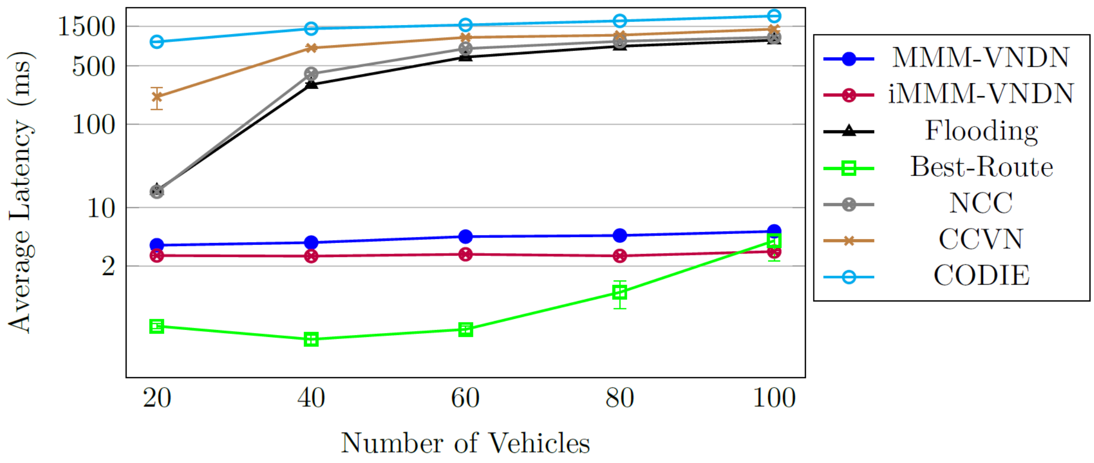



Figure 3.23: Average Latency in Manhattan map for the USLPath for Interest Lifetime of 12 seconds

Figs. 3.24–3.26 shows the average jitter for all strategies. CODIE outperforms our protocols by keeping the jitter very low. iMMM-VNDN and MMM-VNDN perform similar to the previous next hop selection approaches, keeping the jitter above 1ms, compared to the other strategies.

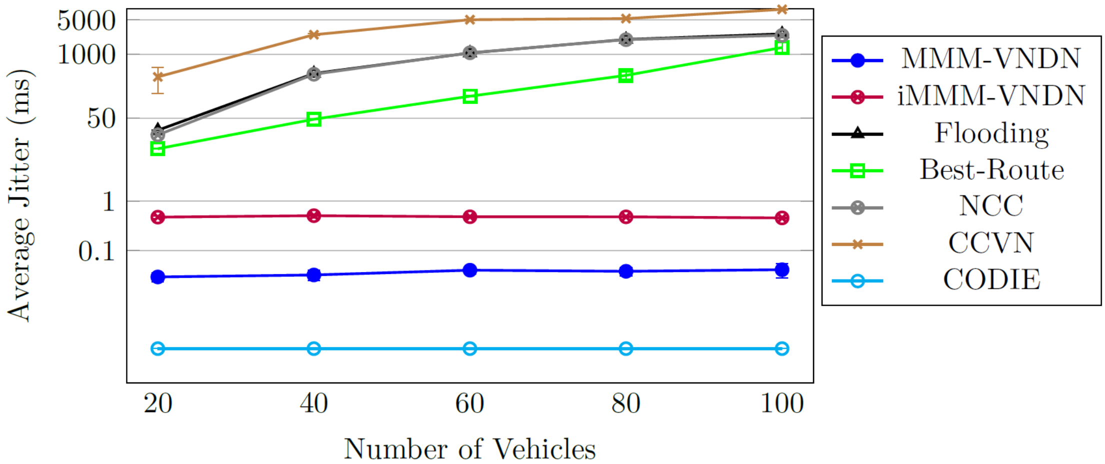



Figure 3.24: Average Jitter in Manhattan map for the USLPath for Interest Lifetime of 4 seconds

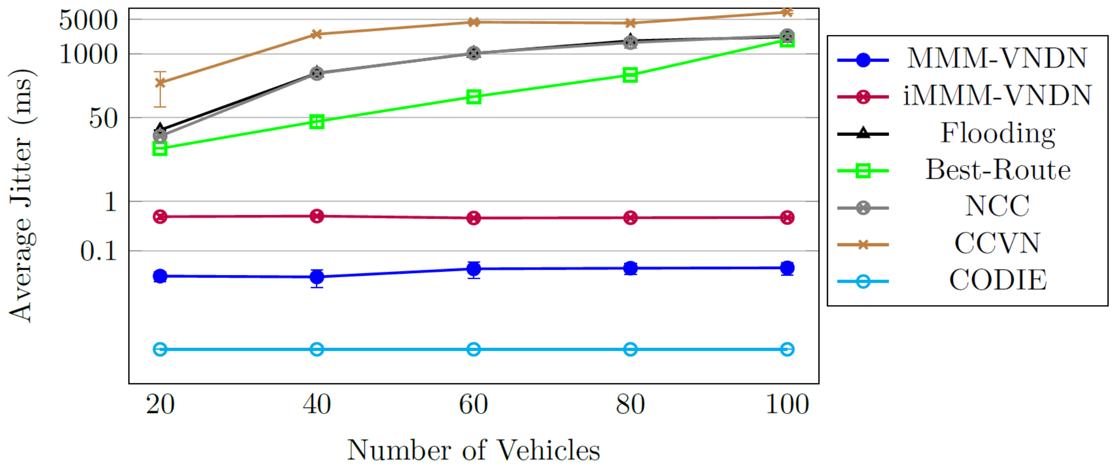



Figure 3.25: Average Jitter in Manhattan map for the USLPath for Interest Lifetime of 8 seconds

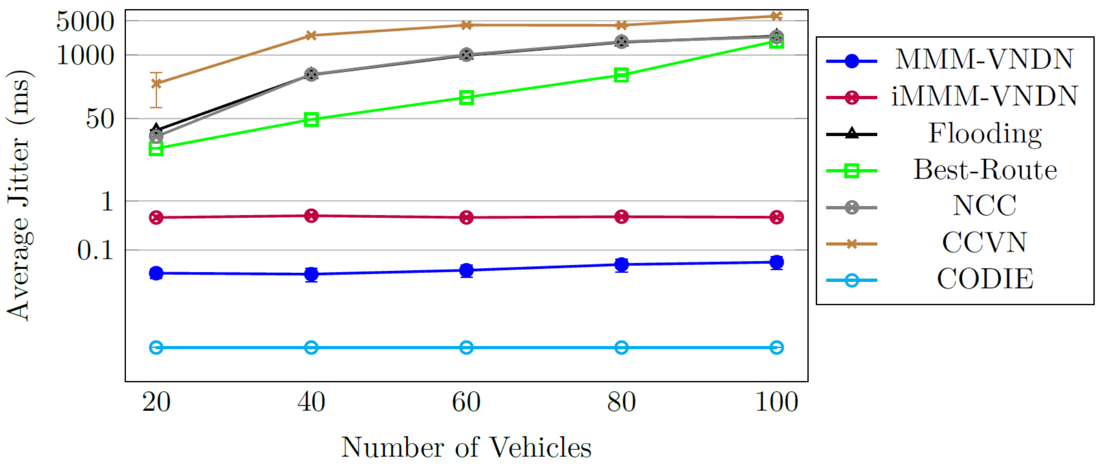



Figure 3.26: Average Jitter in Manhattan map for the USLPath for Interest Lifetime of 12 seconds

Finally, we compare iMMM-VNDN with other algorithms by using the map of the Luxembourg City Center, where we selected an area of 1km x 1km. There, we selected different time slots to extract the mobility traces. Each simulation runs for 149 seconds. We chose to use the traces that have more than 100 nodes to show what happens in a more dense network than the Manhattan map. Since with the Manhattan map iMMM-VNDN performs better than MMM-VNDN, we chose to include only iMMM-VNDN for these experiments in the 3rd next hop selection approach, i.e. when the selection of a next hop is based both on the newest creation time of the entry combined with the lowest latency, compared to the other aforementioned strategies.

Figs. 3.27–3.29 show the ISR for all strategies. The ISR is kept almost the same for iMMM-VNDN independent of the I.L.. It is higher than 94% for a low number of nodes, and it decreases to around 80% if the node density is high.

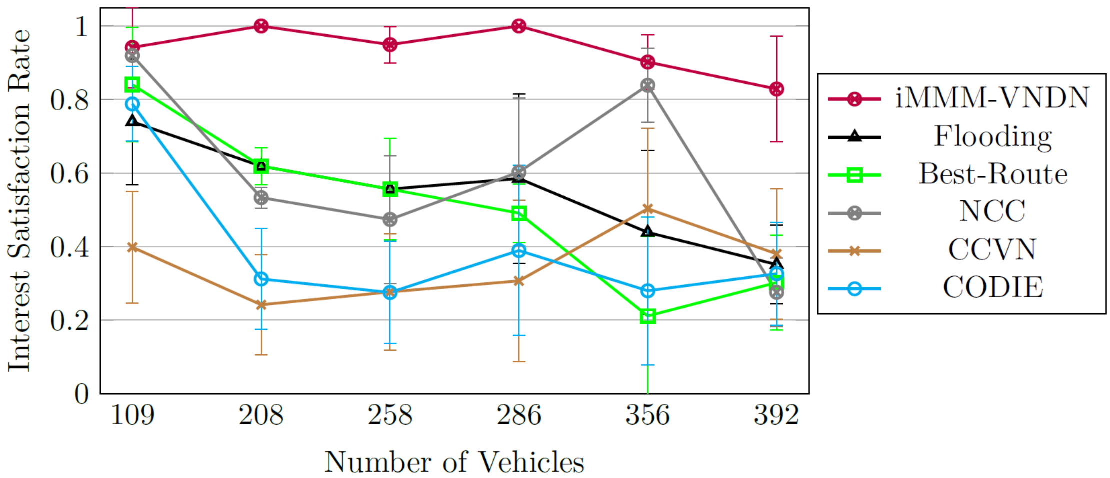



Figure 3.27: ISR in Luxembourg map for the USLPath for Interest Lifetime of 4 seconds

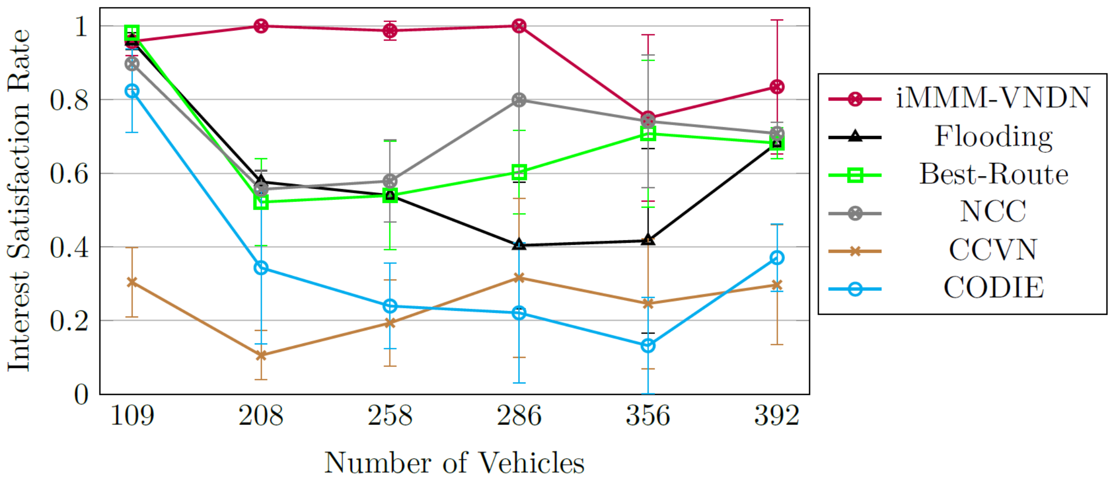



Figure 3.28: ISR in Luxembourg map for the USLPath for Interest Lifetime of 8 seconds

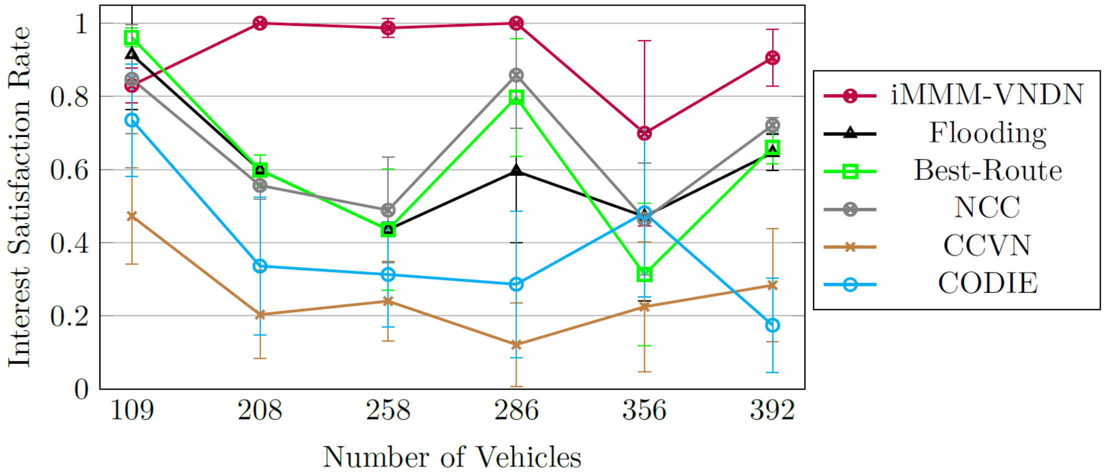



Figure 3.29: ISR in Luxembourg map for the USLPath for Interest Lifetime of 12 seconds

Moreover, iMMM-VNDN outperforms all other approaches considering the average latency, as seen in Figs. 3.30–3.32. In particular, the average latency is stable at 2-4 ms, compared to all other strategies that have a latency bigger than 950 ms.

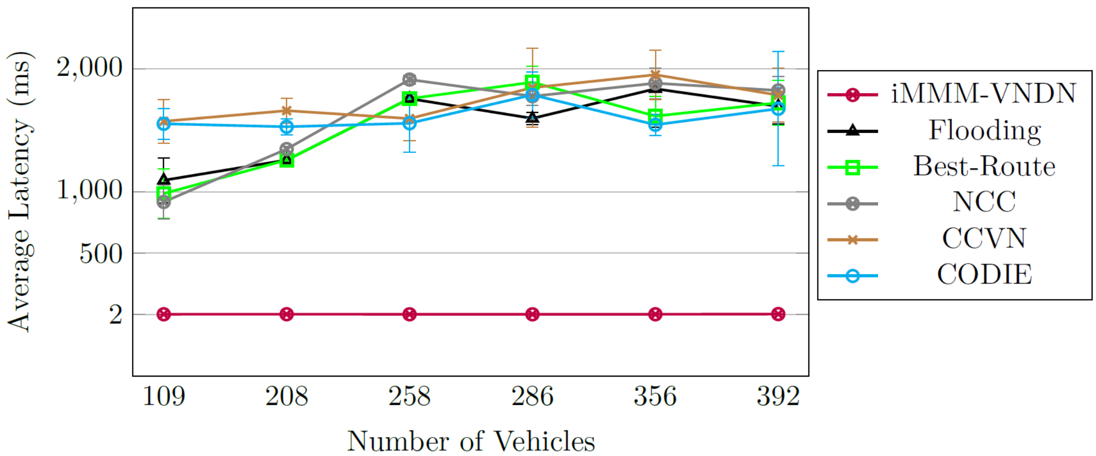



Figure 3.30: Average Latency in Luxembourg map for the USLPath for Interest Lifetime of 4 seconds

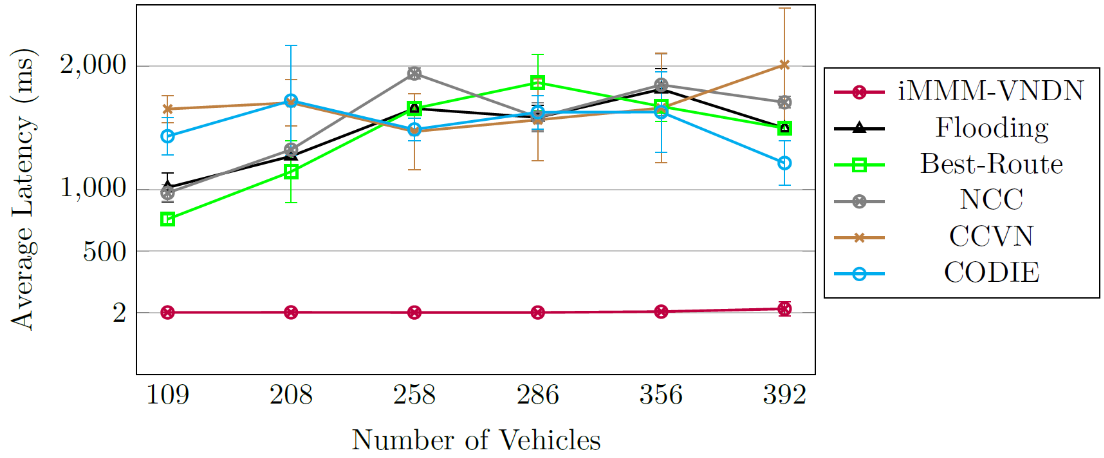



Figure 3.31: Average Latency in Luxembourg map for the USLPath for Interest Lifetime of 8 seconds

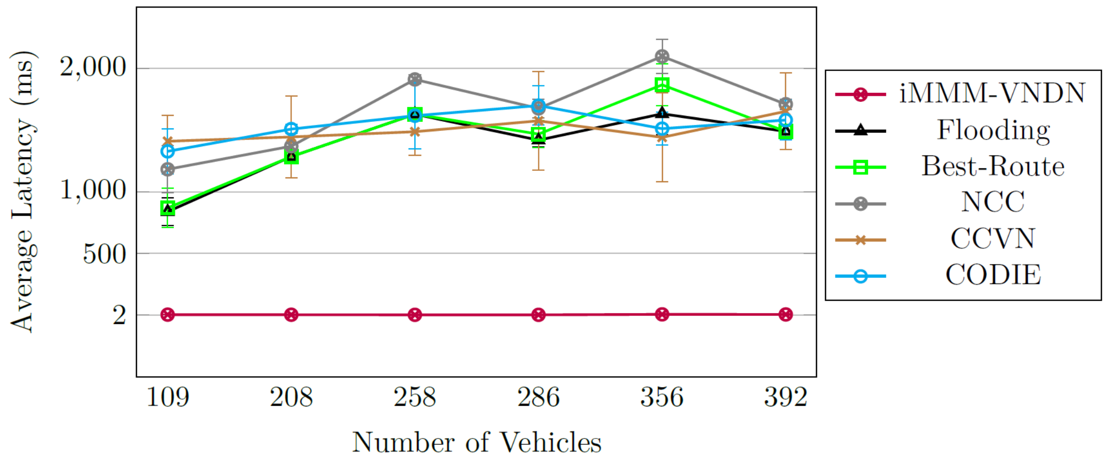



Figure 3.32: Average Latency in Luxembourg map for the USLPath for Interest Lifetime of 12 seconds

We observe in Figs. 3.33–3.35 that the average jitter of iMMM-VNDN is higher than for other approaches, in particular than for the CODIE strategy. This happens, because in a dense network CODIE only manages to retrieve messages that are 1-hop away from the requester node, thus the jitter is non-existent. But, since, in iMMM-VNDN multihop communication is supported, the jitter is increased compared to 1-hop communication algorithms. By creating unicast paths, we can reduce overall message transmissions in the network, and thus avoid collisions that happen in the network. This leads to a more stable environment for message transmissions, since collisions and congestion, in general, are avoided.

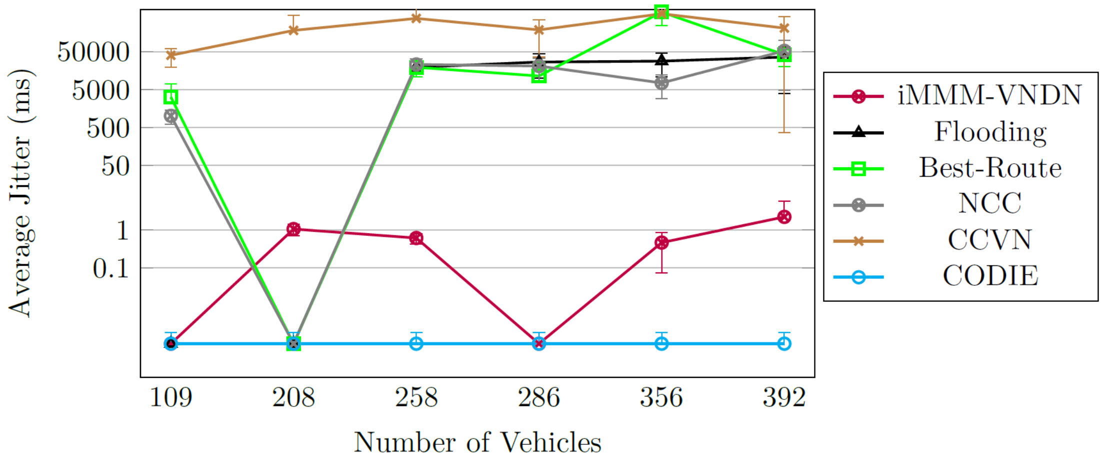



Figure 3.33: Average Jitter in Luxembourg map for the USLPath for Interest Lifetime of 4 seconds

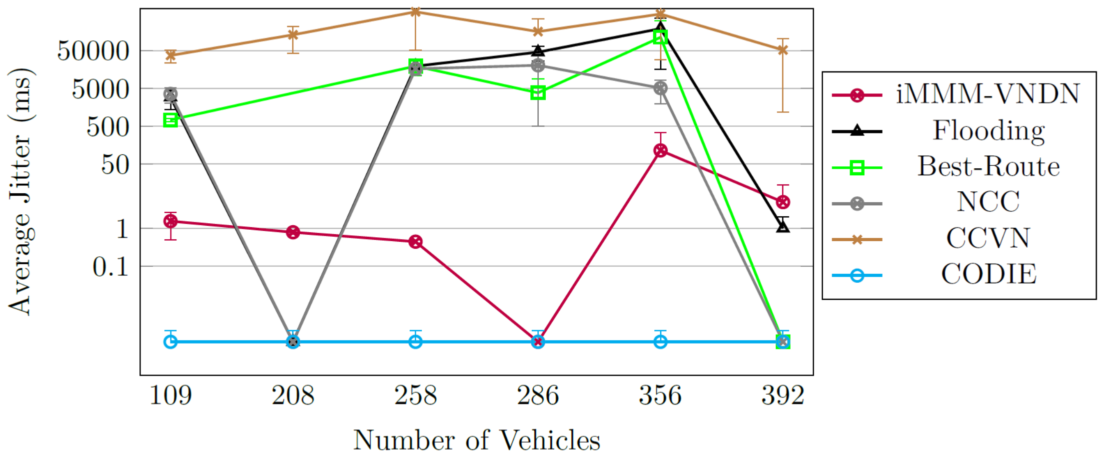



Figure 3.34: Average Jitter in Luxembourg map for the USLPath for Interest Lifetime of 8 seconds

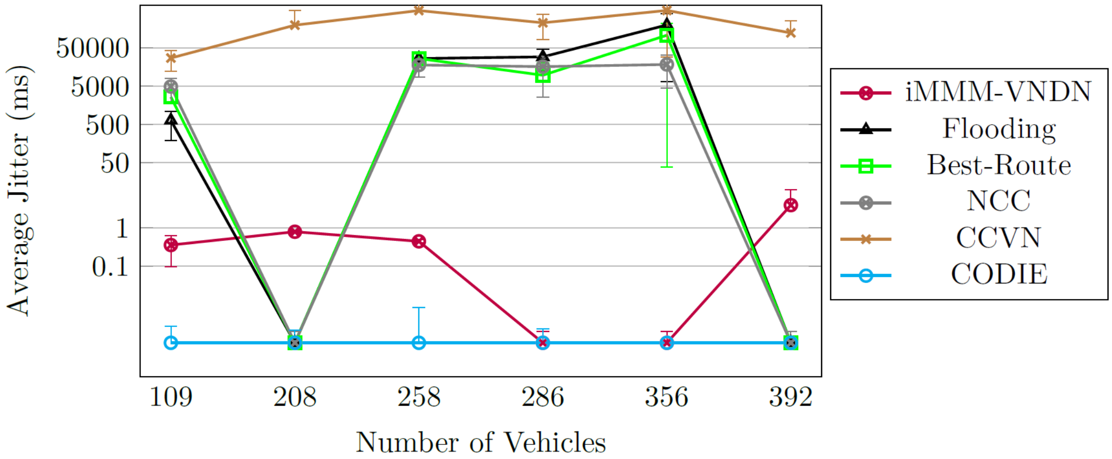



Figure 3.35: Average Jitter in Luxembourg map for the USLPath for Interest Lifetime of 12 seconds

## 3.4 Conclusions

In this Chapter we introduce two routing protocols. The first is called a multihop multipath and multichannel (MMM-VNDN) routing protocol and the second is called improved MMM-VNDN, iMMM-VNDN. Both of these protocols are designed for VANETs using the NDN future Internet architecture. We identified some of the problems that exist in the current architecture of NDN and proposed the corresponding solutions. In particular, we investigated the lack of node identification in NDN and designed a new routing approach that utilizes the MAC addresses of nodes in VANETs. We utilize the new identifiers that are inserted into the nodes of the VANETs and equip each node with multiple interfaces to design two multihop and multipath routing protocols. By this we allow vehicles to create routing entries that we use to unicast messages when possible, and to allow vehicles to transmit and receive messages at the same time.

We also studied different next hop selection techniques, i.e. in which vehicle should a vehicle unicast the message. We choose to route messages uniformly to each next hop of routing entries or based on the latency of a network connection. Finally, we propose that next hop selection should be uniformly and based on the latency of each connection. We compare our algorithms to other state of the art algorithms and with the broadcasting of NDN (every node always broadcasts ). Our results present that the requester node can receive much more Data in less time. Specifically, we show a decrease of up to 12 seconds to the average waiting time, since we do not flood constantly the network with messages, hence, we do not create constant collisions in the communication channel.

The limitations of these algorithms derive from the spreading area of a message together with the periodically broadcasting of requests. The spreading area of a message depends on the antennas that are being used in nodes. This Chapter uses omnidirectional antennas that are installed in nodes, meaning that the power of an antenna is distributed equally around the antenna. Hence, even when unicasting a message, nodes that are in different areas compared to the targeted node will hear the message transmission occupying their channel. Thus, a directional approach is needed to reduce the dissemination area of messages. Moreover, periodically broadcasting is not optimal for content requests. Connections between nodes can be broken much before the next broadcast occurs, hence, nodes can have invalid entries into their routing tables. To avoid this, the interval of broadcasting could be decreased, but then we create again unnecessary traffic into the network. Therefore, an efficient broadcast mechanism should be developed for preventing redundant message broadcasts. We will present our solution for these limitations in the next Chapter.

# 4

# A Geographical Aware Routing Protocol in NDN-VANETs

## 4.1 Introduction

In Chapter 3 we developed an approach for creating FIB entries in the FIB tables of vehicles and we use these entries to unicast messages. We also reduce the broadcast transmissions by broadcasting one message periodically, i.e. in a particular time interval, instead of always broadcasting every message. But, still, as described in Chapter 3, every node periodically broadcasts an Interest message to update its connections and its routing table. But broadcast transmissions may create redundant channel utilization, leading to message collisions (c.f. Section 1.2.1). Moreover, in Chapter 3 we use omnidirectional antennas, meaning that messages are being sent and received in all directions. This leads to messages being delivered to undesired locations (c.f. Section 1.2.2).

In this Chapter, we address **RQ2**, as described in Section 1.2.2, which concerns how to limit the dissemination area of transmitted messages in VANETs when the number of interconnected cars is high. We investigate how we can limit the spreading area of transmitted messages to reduce the collisions on the communication channel. Furthermore, we use a timer-based approach, where each node discovers if its connections (according to its routing table) to other nodes are broken. In particular, we present our **e**nhanced **G**eographical **a**ware **R**outing **P**rotocol, named **eGaRP** [81, 83]. eGaRP focuses on message exchange inside small areas inside a city so that V2V communication is performed without using any infrastructure.

eGaRP does not rely on any infrastructure support. To do so, we assume that a vehicle is autonomous and should perform the necessary actions for meeting its application requirements without infrastructure assistance when this is possible. Considering that VANET services and applications usually concern their surrounding area [74], a vehicle should be able to communicate with its surrounding environment, i.e. with other vehicles in its neighbourhood, so as to collect all the necessary information (Fig. 4.1). Therefore, in a content retrieval process, a vehicle should decide *where*, *how*, and *when* to forward a message.

To perform such a task, we assume that each vehicle runs a navigation application on its On-Board Unit (OBU), which allows the vehicle to know its location at any given time, by using GPS. In addition, we install directional antennas in vehicles to target nodes in a specific direction. By using the GPS coordinates of a vehicle at a given time, and by deploying directional antennas in each vehicle we unicast messages to vehicles that are located at a specific location. The combination of Named Data Networking (NDN) together with directional antennas in vehicles allows retrieving content based on its name and sending requests to a specific location, leaving nodes that do not need to participate in the content exchange unoccupied. Moreover, each vehicle is equipped with directional antennas that cover 360° are around the vehicle, and these antennas are used only to transmit a message towards a direction. Since we allow 360° coverage area on a vehicle, we can guarantee that the message will be received in any of the antennas of a vehicle. In particular, the contributions of this Chapter are summarized as follows:

- We introduce in detail our proposed enhanced Geographical aware Routing Protocol (eGaRP) to support multihop V2V communication in vehicles for content retrieval.
- We install directional antennas in each vehicle to support directional forwarding and to reduce the dissemination area of transmission of a message.

- We unicast requests for a particular content object, after a path from the requester to the content source has been established.

- To support vehicle mobility and path breaks we use a timer-contention-based forwarding mechanism. In particular, in our approach, a vehicle decides *when* a message path to the content source is broken. Then, it is responsible for unicasting the Interest via another path (when a unicast path is available) to the content source. Our algorithm allows vehicles to retrieve requested content quickly, as it reduces the number of overall messages of the network, by sending messages to a specific location.

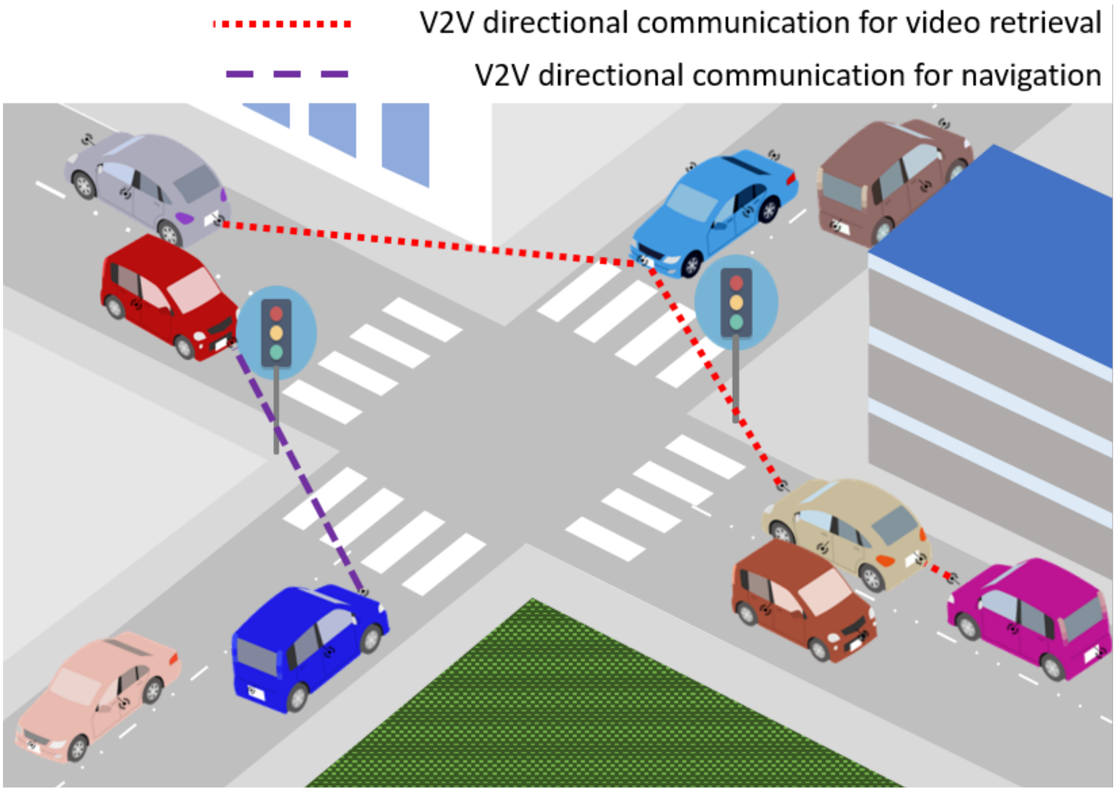



Figure 4.1: V2V directional communication using directional antennas.

Fig. 4.1 presents our idea, where vehicles in an intersection retrieve different content objects from different paths. Each path is created only by vehicles that need to participate in a content exchange mechanism, leaving other vehicles unoccupied to perform other tasks. Every vehicle has 4 directional antennas installed, making the communication possible in particular directions. For instance, three vehicles are participating in a video retrieval process, and each of the vehicles uses 1 directional antenna. At the same time, 2 other vehicles communicate for the execution of a navigation application, again with each vehicle using 1 directional antenna.

The rest of this Chapter is structured as follows: In Section 4.2 we present our algorithm for content retrieval via V2V and in Section 4.3 we evaluate our work. We conclude this Chapter in Section 4.4.

## 4.2 System Model

Although cellular infrastructure is already deployed in many cities and will be upgraded to 5G technology, a vehicle cannot rely on accessing information via cellular infrastructure only, because infrastructure can fail unexpectedly. A vehicle should be autonomous to perform actions like driving, accessing vehicle information, accessing community information, downloading or uploading infotainment content to other vehicles, and also retrieving and broadcasting, when necessary, emergency messages. In this Chapter, we focus on making infotainment content accessible to vehicles when this content is requested and is available within a particular geographical area. We do not rely on any infrastructure, but we rather let a vehicle create routes and then decide where and how to send a message.

### 4.2.1 Forwarding Support

One of the main problems in VANETs is the intermittent connectivity between vehicles. There are many ways to establish a message route from the source node to the destination node. To solve this problem and to update all routes between vehicles the literature proposes either infrastructure support [148], always broadcasting each request [28], limit the forwarding area based on geographical coordinates in messages [72], or a combination of the above. In our previous work named iMMM-VNDN (c.f. Chapter 3) [84] we proposed a combination of broadcast and unicast to support intermittent connectivity and to establish paths between vehicles. The main focus of this Chapter is to reduce the resources that are required during the exchange of a content object. To reduce required resources we need to:

- Reduce redundant messages that are broadcast, when a path has been established.

- Use unicast transmissions, when content can be retrieved by established paths.
- Limit the message spreading area, i.e. the direction of the message that is transmitted, to use as little resources as possible.

To perform the above actions we use as in [84] MAC addresses to identify nodes between source and destination vehicles and to create unicast paths based on these MAC addresses. To limit the dissemination area of a message we propose to use directional antennas to target at once the geographical location that a message should be sent. When a message is forwarded into a particular direction, which is denoted by the directional antenna that is being used to forward the message, other vehicles outside of the area that is covered by the directional antenna remain unaffected by this message exchange and can perform different actions.

### 4.2.2 Placement of Directional Antennas

Vehicles are equipped with directional antennas. Each antenna points into a different direction to target different locations. The direction of each antenna is associated with the number of antennas of a vehicle. Therefore, if a vehicle contains $N \in \mathbb{Z}^+$ antennas, each antenna $A_i, i \in \{1..N\}$ will have beamwidth $B$:

$$B = \frac{360°}{N} \tag{4.1}$$

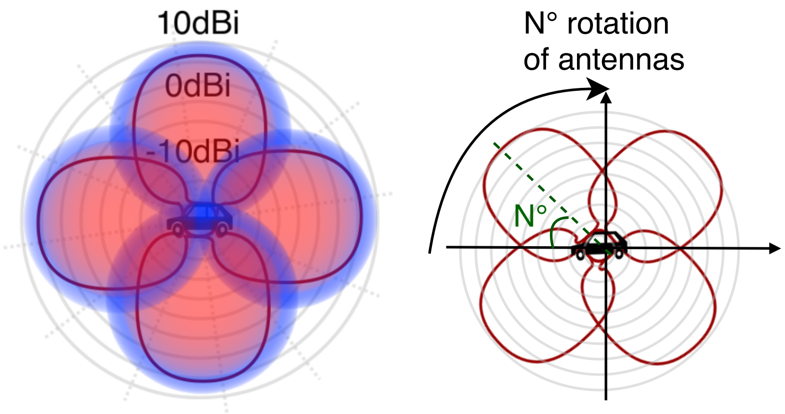
and the pointing $P_{A_i}$ of an antenna $A_i$, i.e. where the antenna radiates and receives its greatest power, will be:

$$P_{A_i} = (i-1)B + B/2, \ \forall i \in \{1..N\} \tag{4.2}$$

Thus, if, e.g. a vehicle consists of 4 directional antennas we place the antennas in the vehicle as shown in Fig. 4.2b.

### 4.2.3 Rotating Antennas

In this study, we suggest using directional antennas to send and retrieve messages to and from a particular location. One main problem is the radiation pattern of the directional antennas. Fig. 4.2a shows a vehicle that is equipped with 4 directional




(a) (b)

Figure 4.2: Example with 4 patch antennas installed in a vehicle. (a) Placement of 4 patch antennas in a vehicle as described in [59]. Blue indicates low gain and red high gain of an antenna. (b) Rotation of antennas by N°.

antennas, in particular patch antennas [59]. A patch antenna is an example of a directional antenna, whose radiation pattern is characterized by a single main lobe of moderate beamwidth. The beamwidth of a patch antenna can be manipulated to produce a higher or lower gain in particular areas, depending on the requirements [46]. Assuming that antennas are installed on a vehicle and that their effective gains are not coinciding with each other, some areas around the vehicle are not covered with a signal, i.e. the gain of the antennas is small. Thus, nothing can be received, as in Fig. 4.2a, in areas denoted outside of the antenna lines (by blue). To avoid such an issue we propose to mechanically rotate the antennas by an appropriate angle. Each antenna covers a particular area, which is derived according to the antenna type, its beamwidth, and its radiation pattern. To send a message a particular directional antenna is selected. This selection is based on a node's location and on the next hop's location (the selection process is described in Section 4.2.5). We rotate the selected antenna to point into a particular direction to improve the antenna gain and to allow a better connection to the vehicle that we want to send a message to. Then, we rotate all antennas of the vehicle by the same angle as we turn the selected antenna, to guarantee that the antennas of the vehicle will continue to have a 360° coverage area, as shown in Fig. 4.2b.

Table 4.1: NDN DATA STRUCTURES. NEW FIELDS ARE DENOTED IN ITALIC

(a)

| PIT Table |
|---|
| Prefix |
| Face (MAC Address) |
| Expiration timer |
| *Current geographical coordinates* |
| *Creation time* |

(b)

| FIB Table |
|---|
| Prefix |
| Face (MAC Address) |
| Latency |
| *Current geographical coordinates* |
| *Hop count* |

(c)

| NDN messages additional info |
|---|
| *Current geographical coordinates* |

### 4.2.4 Changes to NDN Data Structures

Each directional antenna that is placed on a vehicle is attached to a network interface card. Therefore, a vehicle has many interfaces, each with a different MAC address. As in [84] to exchange content in a VANET we target nodes based on their MAC addresses to perform a unicast transmission. The main goal is to find a path between the node that requests content (requester) and the node that holds the content (content source) by using available information that is retrieved by surrounding vehicles.

We assume that each node has a GPS device installed on its OBU. Hence, it knows its current position. The navigation device gives a route suggestion to the driver using GPS and, thus, the vehicle can extract its current geographical coordinates at any given timestamp from the GPS device. We extended the NDN stack of the Interest and Data message by including the node's current geographical coordinates according to the GPS device. The new field in the NDN message structure can be seen in Table 4.1c.

Every vehicle contains only the traditional NDN data structures, i.e. PIT, FIB, and CS. In our previous work [84] we included a new field in the PIT and FIB tables, the MAC address. In this study, we add new fields in these data structures to include the current geographical coordinates of the node. The structure of the PIT and the FIB tables can be seen in Table 4.1a and 4.1b, respectively. In the PIT we added a new field, the creation time, which indicates the time that the PIT entry was created. In the FIB table, we added a hop count to assist us to choose the routing entry with the lowest hop count to unicast an Interest message.

### 4.2.5 Route Discovery and Forwarding

To retrieve content, we create at least one path between the requester and the content source. To create this path, we create unicast routes between two vehicles and then we connect these routes hop by hop to send the message to the content source via multiple hops (if needed).

#### Interest Processing

Content retrieval starts by sending Interest messages into the network. As in traditional NDN, a node checks its FIB table to identify the next hop(s) to send the Interest message. When a requester wants to send the first Interest message for the content to be retrieved, the FIB table is empty. Therefore, the node broadcasts the Interest message through all its available directional antennas. When the requester broadcasts the Interest message through one of its interfaces, it includes the corresponding MAC address of the interface [84] and its current geographical coordinates according to its GPS device. The extracted geographical coordinates from every vehicle's GPS device have $\pm 7.1$m accuracy [166].

When the first Interest, transmitted via broadcast, arrives at a node, as in traditional NDN, the node checks its PIT to identify previous Interests that the node has already forwarded. If the node has forwarded the Interest before, it extracts from the Interest the MAC address and geographical coordinates, it enters this information into the PIT, and discards the Interest. If the node has not forwarded the Interest before, it enters the same information into the PIT. Then, it updates the Interest to contain its own MAC address and current position and broadcasts the Interest through all its antennas into the network. This continues until a node with the content (content source) receives the Interest.

#### Data Processing

When the content source receives the Interest, it responds with a Data message. This Data is unicast to the node that the Interest arrived from, by using the node's target MAC address [84]. Since the Data message is unicast, the content source should choose an appropriate antenna for transmitting the Data. The content source extracts the current geographical coordinates of the PIT entry, i.e. the position of the node that the content source wants to send the Data message to. Then, the content source calculates the angle (by using simple trigonometric functions) between its own position (that is known from its own GPS device) to the node that should receive the Data message, by taking into account the other node's position as stored in the PIT. After the angle has been calculated the content source selects the appropriate antenna that covers this angle with its direction pattern. In Fig. 4.3 $V_A$ wants to send a message to $V_B$. It calculates the angle $\phi$, taking into account $(X_B, Y_B)$ and $(X_A, Y_A)$.

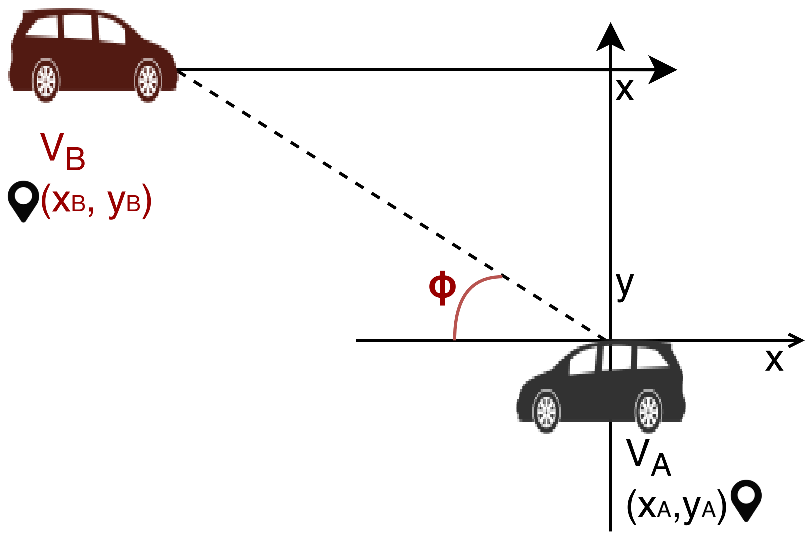



Figure 4.3: Calculation of angle $\phi$ for antenna selection.

After the appropriate interface with the correct antenna has been chosen, the content source updates the Data message to include the MAC address of the chosen interface and its own position. Then, the content source rotates the selected antenna to point to the calculated angle and unicasts the Data through the selected interface and, thus, the appropriate directional antenna to the next node.

Intermediate nodes receiving the Data message check the MAC address that is included in it to determine if the Data is meant for them [84]. If not, the message is discarded. Otherwise, the process of a Data message in an intermediate node is as in Fig. 4.4 and it is the same as in a content source. In addition, intermediate nodes increment the hop count field in the Data message by one before unicasting it to the next hop.

When the first Interest (that is broadcast from all nodes until it reaches the content source) is satisfied, i.e. the corresponding Data message has been received by the

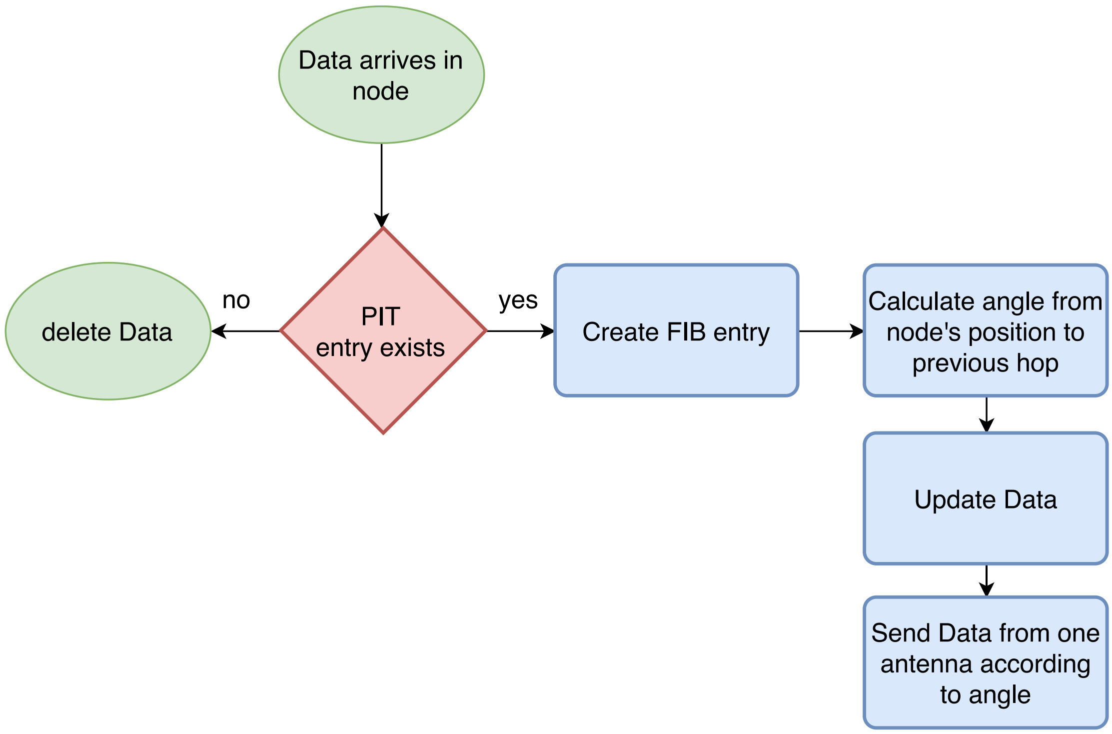



Figure 4.4: Processing of a Data message in intermediate nodes.

requester, the FIB table of the nodes that participated in the first Data retrieval contains routes to other nodes. Thus, when the requester sends the second Interest message, it selects a route to the content source from its FIB. The requester selects the FIB entry with the lowest hop count and calculates an angle between its own position (geographical coordinates that are extracted from the GPS device) and the geographical coordinates of the selected FIB entry. After, the requester updates the Interest message to include its own MAC address and geographical coordinates (according to the GPS device). Having the angle and knowing the position of its own antennas, i.e. beamwidth and coverage area of each antenna, the node selects the appropriate network interface, rotates its antenna (the rotation takes from 50-100 ms) and unicasts the Interest (by using the MAC address of the target node that is extracted from the FIB entry) through this network interface (and thus, a specific directional antenna). When the target node (next hop) receives the unicast Interest message, it checks the Interest's MAC address to identify if the Interest is for meant for it as in [84]. If not, the Interest is discarded. Otherwise, the process continues as in the requester, by inserting an entry in the PIT, checking the FIB to identify a route to a next node, updating the Interest with its own information, calculating the angle between

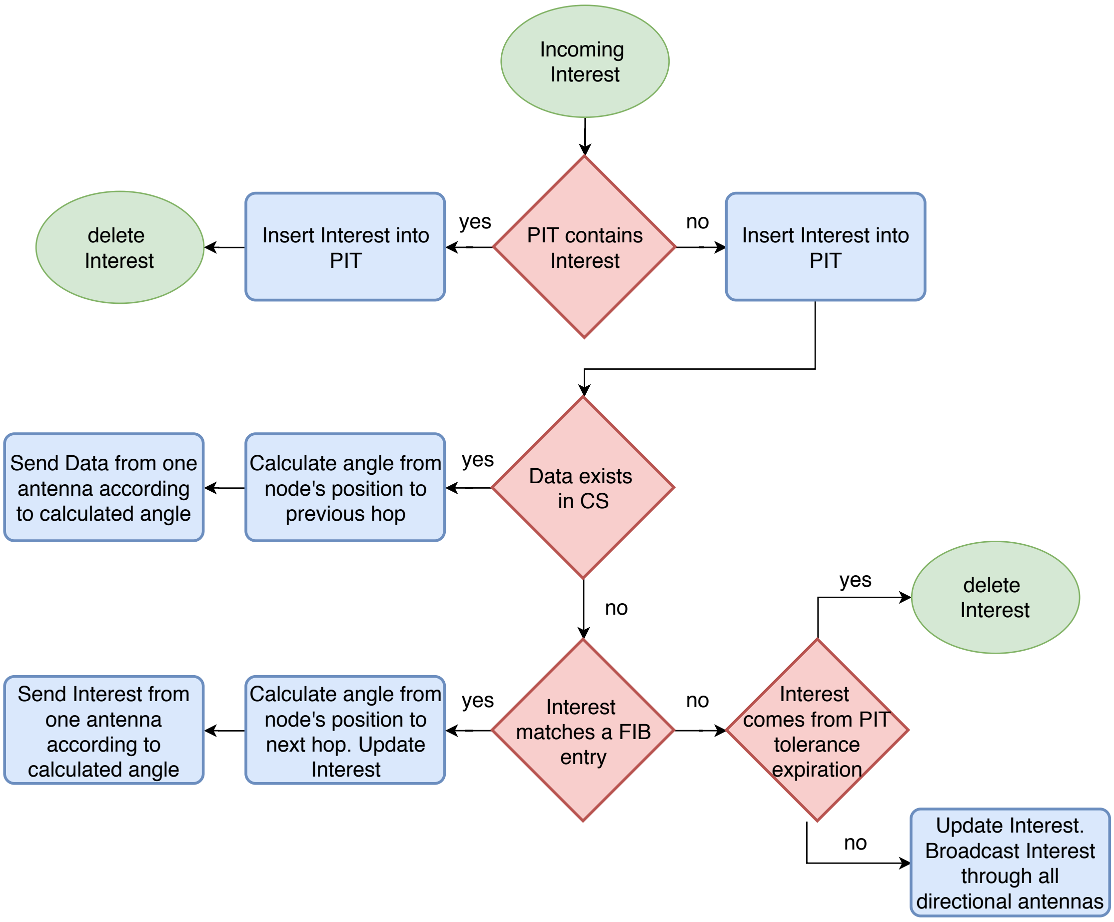



Figure 4.5: Processing of an Interest message in intermediate nodes.

itself and the target node (according to its own geographical coordinates and the geographical coordinates from the selected FIB entry), rotating the selected antenna, and unicasting the Interest to the next node (Fig. 4.5). This process continues at all intermediate nodes until the Interest arrives at the content source. The content source then processes the Interest and responds with the proper Data message by unicasting the Data, following the same procedure as it did before (Fig. 4.4).

### 4.2.6 Mobility Support by Route Rediscovery and Duplicate Suppression

Since VANETs are characterized by intermittent connectivity due to mobility, the routes between vehicles might break unexpectedly. This leads to path breaks resulting

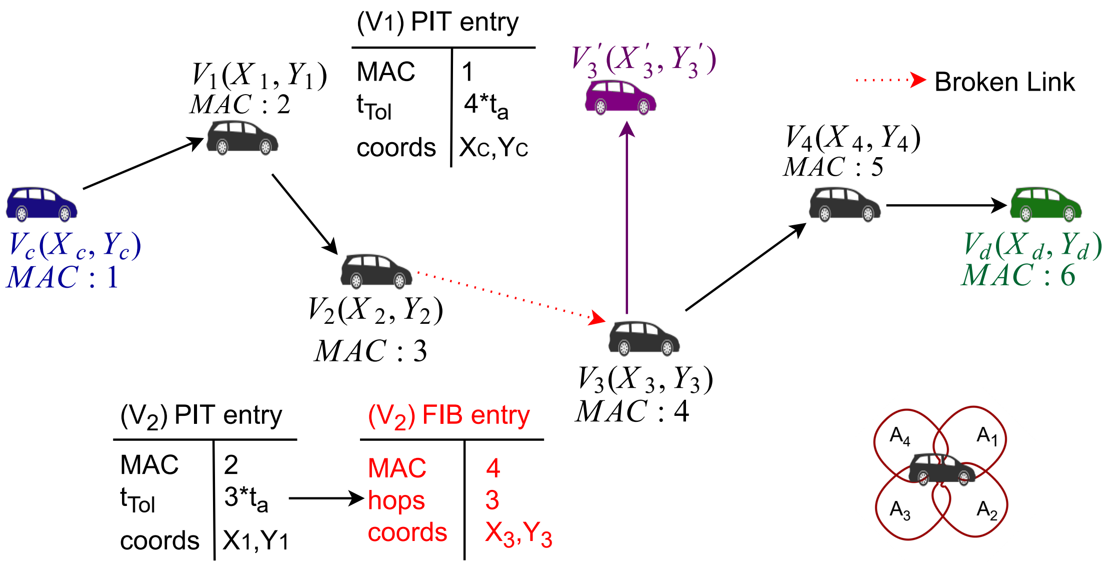



Figure 4.6: Example of a V2V topology. $V_c$ requests Data and $V_d$ holds the Data. Vehicles in black are intermediate nodes for Data retrieval.

in unsatisfied content. For instance, in Fig. 4.6 we assume that $V_c$ is the vehicle requesting content and $V_d$ is the vehicle that holds the content. $V_c$ sends Interest messages according to its FIB to a next node $V_1$ using antenna $A_1$. Then, the intermediate vehicle $V_1$ sends a message to a next hop $V_2$ and $V_2$ sends it to $V_3$. $V_2$ unicasts this message by selecting the directional antenna $A_2$ and includes the MAC address of the next hop in the message. But due to mobility, the next node ($V_3$) can travel to a new geographical position and does not receive this unicast message. Therefore, the requests of $V_c$ remain unsatisfied.

There are several ways to deal with path breaks. The first, which is proposed in the traditional NDN scheme, is that only the node that requests the content starts retransmitting the Interest if the Interest is unsatisfied, i.e. if no Data message has been delivered to the requester as a response to the Interest transmission. In that case in Fig. 4.6 only $V_c$ will retransmit the Interest if the path from $V_c$ to $V_d$ breaks. Another way is that nodes that receive and forward the Data (as a response to an Interest) will be responsible for guaranteeing that the Data has been delivered to the next hop node (e.g. by exchanging of ACKs from the node that sends the Data to the node that receives the Data). For instance in Fig. 4.6, if $V_3$ moves from position $X_3, Y_3$ to $X_3', Y_3'$, then $V_3$ is responsible for ensuring that $V_2$ receives the transmitted Data message.

In this work, we let the decision of detecting a path break and re-establishing new paths to the content source to each node that forwarded an Interest message. For instance, in Fig. 4.6 $V_2$ is now responsible for finding a unicast path that leads to $V_d$. Our algorithm uses a timer-contention-based forwarding algorithm ( [66]), where each node decides individually when (by using timers) and where to forward a message to find a route that leads to the content source. The idea of a timer-contention-based forwarding algorithm is that each node sets a timer, and the node performs some actions when this timer expires.

In eGaRP, we use the idea of timer-contention-based forwarding. In particular, when a node unicasts an Interest, it chooses the FIB entry with the lowest hop count. Then, the node enters the Interest's information into the PIT and it also associates this PIT entry to the corresponding FIB entry that has been chosen for forwarding.

This newly created PIT entry contains an expiration timer, i.e. how long a vehicle should wait for a Data message. We define this expiration timer as the *tolerance* of an Interest in a vehicle $t_{Tol}$ (for simplicity we just call it vehicle's tolerance and assume that it is about the same PIT entry).

Since we associated the PIT entry to the FIB entry, we extract the hop count field $h$, $h \in \mathbb{N}$, i.e. how far away the content source is in terms of vehicles, from the FIB. Then, we define $t_\alpha$ as the minimum time that a node has to wait for a Data message to be retrieved. Finally, we define $t_{Tol} = ht_\alpha$, with respect that $t + t_{Tol} \leq t_{PITcreation} + t_{INTlif}$ and $t_\alpha \leq t_{Tol} \leq t_{INTlif}$, where $t$ is the current timestamp, $t_{PITcreation}$ denotes the time when the PIT entry was created, and $t_{INTlif}$ is the Interest lifetime (when the Interest will time out). In this case it is always $t = t_{PITcreation}$. The value $t_\alpha$ can be defined according to the requirements of the application in the requester. A high $t_\alpha$ means a more delay-tolerant application than a lower $t_\alpha$.

For instance, in Fig. 4.6 $V_2$ selects from its FIB MAC address 4 to unicast a message and then enters to its PIT a pointer to the selected FIB entry. Then, it extracts from the FIB the hop count of the selected FIB entry and defines in its PIT the $t_{Tol} = 3t_\alpha$.

When the tolerance of a node expires, i.e. a Data message is not received by the node in the accepted time limit $t_{Tol} = ht_\alpha$, then the PIT entry is considered unsatisfied and actions should be taken to satisfy this PIT entry. First, the node deletes the corresponding FIB entry that was selected for forwarding and associated with the expired PIT entry, because it assumes that this route is now invalid, i.e. this FIB entry cannot satisfy its Interests because either the path is broken or there exists network traffic along this path. After, the node searches its FIB table to identify another FIB entry to unicast the same Interest. If there is a FIB entry, then the node retransmits the Interest according to the newly selected FIB entry, as described in Section 4.2.5. If there is no FIB entry, the node does not have a unicast route to the content source, hence it cannot unicast the Interest. In this case, since our goal is to reduce the broadcast transmissions to reduce the number of messages that vehicles receive and to occupy as little as possible resources, we do not broadcast an Interest from an intermediate node. Instead, we enter a new $t'_{Tol} = t_{INTlif}$ to the PIT entry, which is the maximum tolerance. If the new $t'_{Tol}$ expires, both the Interest and the tolerance will expire. We, then, delete this PIT entry. So, if vehicles in a path do not have any more unicast routes to the content source, we let only the requester to retransmit the expired Interest. When a retransmitted Interest from the requester arrives at an intermediate node that does not have FIB entries (i.e. no unicast paths to the content source), then the intermediate node will broadcast the Interest as explained in Section 4.2.5 (Fig. 4.5). In Fig. 4.6 no content is retrieved after $3t_\alpha$, because $V_3$'s position has changed to $X'_3, Y'_3$. The selected (red) FIB entry of $V_2$ will be deleted, and since $V_2$ does not have more FIB entries, it will redefine its $t_{Tol} = t_{INTlif}$ starting from that timestamp.

As mentioned before we define the tolerance as $t_{Tol} = ht_\alpha$. $t_\alpha$ is a constant and $h$ is the distance in terms of hop counts to the content source. Nodes that are closer (in terms of hop count) to the content source will have a lower $h$ than others that will be further away, i.e. that will have a higher $h$. Therefore, the tolerance of nodes closer to the content source will be shorter, compared to nodes that are further away from the content source. These nodes will forward Interests to discover new routes sooner than nodes that are further away from the content source (in terms of hop count).

If the node that is closer to the content source in terms of hop count does not receive a Data message, it retransmits the Interest. Likely, some intermediate nodes that are also further away from the content source will not receive the Data, so when their tolerance expires, they will retransmit the same Interest. To avoid transmitting the same Interest from many connected nodes at almost the same time, we change the vehicle's tolerance, when the same Interest is overheard by the vehicle. In particular, every time that an Interest transmission is overheard by a node, even if the Interest is not meant for this node, the node checks its PIT to identify if it has forwarded the message before. If it has not sent the message before and the Interest is not meant for this node, then it discards the Interest. But, if the node transmitted the Interest before, then there will be a PIT entry that will have an expiration timer, i.e. the vehicle has already defined a tolerance. Then, we reset the tolerance for this PIT entry to start from the time point the node received the Interest unless this causes the tolerance to exceed the lifetime of the Interest. This is how we perform duplicate suppression to avoid transmitting the same Interest messages in the network when they are redundant. For instance in Fig. 4.6, $V_2$ will have $t_{Tol} = 3t_\alpha$. Because the link from $V_2$ to $V_3$ breaks, i.e. the PIT entry timer expires, the PIT entry is considered unsatisfied. Assuming that $V_2$ has another FIB entry, $V_2$ will unicast the Interest message to another node. $V_1$ has also an entry in its PIT with $t_{Tol} = 4t_\alpha$. When $V_1$ receives the re-unicast Interest from $V_2$ after $t = 3t_\alpha$, the time that remains until its $t_{Tol}$ expires is $t_\alpha$. So, at time $t = 3t_\alpha$, $V_1$ will reset its PIT entry's expiration timer to start again ($t_{Tol} = 4t_\alpha$).

## 4.3 Performance Evaluation

### 4.3.1 Simulation Environment

#### Simulation Scenario

We use the LUST scenario [48] that is based on real traffic of the Luxembourg city over 24 hours. We extract an area of 1km x 1km in the city centre and choose 150 seconds during the rush hours (6 pm) to run our algorithm. Then, in this area, we choose two nodes that are crossing the whole area in these 150 seconds. One node is the requester node and one is the content source.

#### Simulation Parameters

We evaluated our protocol by using the OMNET++ network simulator with the vehicular framework VEINS to support vehicle communication and SUMO to support mobility. We changed our simulator from the ns-3 to OMNET++, since ns-3 does not support directional antennas together with the IEEE 802.11p protocol. Therefore, in this thesis, when directional antennas are used, we use the OMNET++ simulator, and when only omnidirectional antennas are used we use the ns-3 simulator.

Every vehicle consists of 4 interfaces and, therefore, 4 directional antennas, each pointing into a different direction. We choose 4 interfaces as a basic configuration of antennas in the vehicle, although installing more than 4 antennas is a viable option and is studied in Chapter 6. Based on Eq. 4.1 each of the antennas will have beamwidth $B = 90°$. According to Eq. 4.2 the first antenna will point to 45°, the second to 135°, the third to 225°, and the fourth to 315° (Fig. 4.2). In the presented results the geographical coordinates that are extracted from the GPS devices running in vehicles contain an error of $\pm 7.1m$ to support realistic GPS coordinates. A synopsis of the simulation parameters is shown in Table 4.2.

Table 4.2: SIMULATION PARAMETERS

| **Parameter** | **Value** |
|---|---|
| Channel Frequency | 5.890e9 Hz |
| Sensitivity | -89 dBm |
| Transmission Power | 20mW |
| Propagation loss model | Two Ray |
| Bit Rate | 6Mbps |
| Phy Model | IEEE 802.11p |
| Number of interfaces (antennas) | 4 |
| Number of vehicles | 158 |
| Average Vehicle Speed | 20-30 m/s [48] |
| Area | $1km^2$ |
| Interest interval | 1 s |
| Simulation time | 150 s |
| GPS accuracy | ± 7.1 m [166] |

The parameters chosen for the physical and the MAC layer are proposed in the IEEE 802.11p. Since we choose a small area from the LUST scenario [48], we measure the number of nodes entering this area during the 150 seconds of simulation time.

### Simulation Metrics

We divide the performance metrics into two categories. First, the following metrics concern the requester node and its traffic in its application layer only:

- *Number of Delivered Data* shows the number of received Data messages.
- *Average Latency* shows the mean delay of the received Data messages. This metric denotes the average time between the transmission of an Interest message to the reception of the corresponding Data message.

- *Interest Retransmissions* denote how many times the node has to resend an unsatisfied Interest message. An Interest message is unsatisfied, if no Data message is received as an answer to it, or if it has timed out.

Moreover, to evaluate our algorithm we included additional metrics that concern the whole network, i.e. all nodes that pass through the selected area during the 150 seconds. For all nodes we count their incoming and outgoing messages and display the average number for one node in the NDN layer:

- *Interests Received*: The average number of Interests a node has received.
- *Sent Interest Unicasts*: The average number of Interests a node has unicast to another node.
- *Sent Interest Broadcasts*: The average number of Interests a node has broadcast to the network.
- *Data Received*: The average number of Data messages a node has received.
- *Sent Data Unicasts*: The average number of Data messages a node has unicast.

We additionally display on the MAC layer:

- *Received Broadcast Packets*: The average number of broadcast messages a node has received in one network interface card.
- *Received Unicast Packets*: The average number of unicast messages a node has received in one network interface card.

The above metrics describe the main characteristics that we consider important to the V2V communication in a VANET for an infotainment application. ISR is a core metric, but it is insufficient, when a VANET application has delay constraints. Hence, latency is measured in our experiments. Moreover, we measure how many Interests are retransmitted, to analyse how much traffic the requester node sends to the network. Furthermore, we measure the average number of packets (both Interest and Data messages) a node receives and sends to further analyse the impact of $t_\alpha$, in the network.

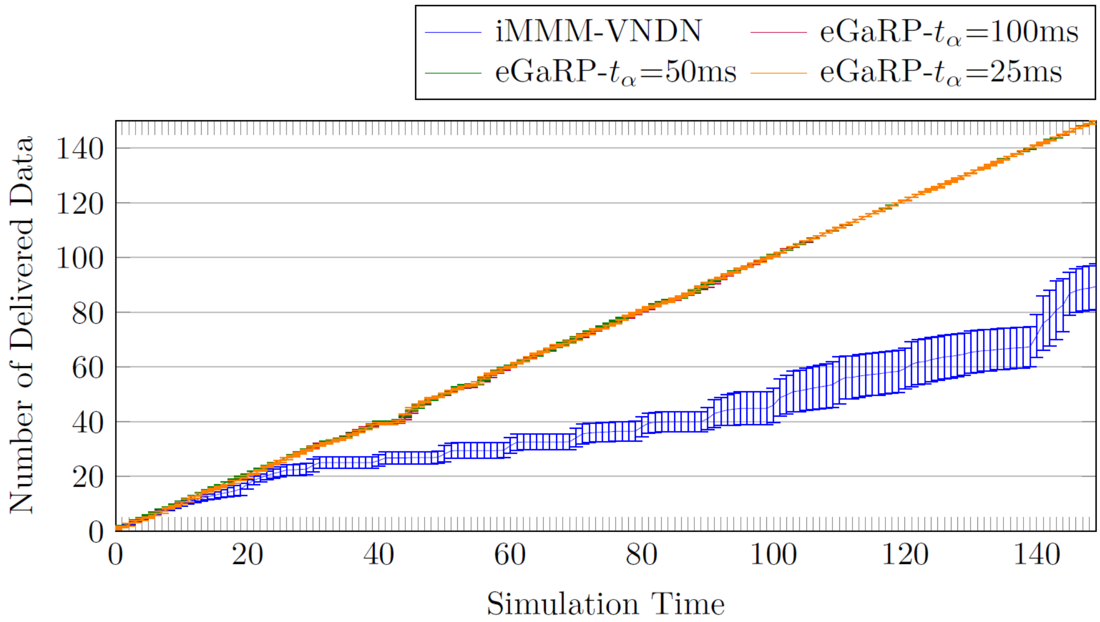



Figure 4.7: Number of Delivered Data in the application layer of the requester node.

### 4.3.2 Simulation Results

We compare our algorithm, eGaRP, using different $t_\alpha$, $t_\alpha = 100ms$, $t_\alpha = 50ms$ and $t_\alpha = 25ms$ with our previous V2V algorithm iMMM-VNDN [84], as described in Chapter 3. We reimplemented iMMM-VNDN in the OMNET++ simulator to guarantee a fair comparison of our results. In [84] all vehicles are equipped with 4 interfaces, and in each interface one omnidirectional antenna is attached. In all experiments, we define $t_{INTlif} = 1s$. Figs. 4.7-4.9 present the results of the application layer of the requester node as a function of the simulation time.

In particular, Fig. 4.7 presents the number of delivered Data messages in the application on the requester node. We see that eGaRP outperforms iMMM-VNDN [84] since we receive all the requested Data messages. The requester node sends 1 Interest/second from the application and receives 1 Data message almost every second, in the case of eGaRP.

Fig. 4.8 presents the average latency of the application during the Data retrieval process. We observe that $t_\alpha$ affects the application delay. When $t_\alpha$ is low, the latency is also low. Intermediate nodes retransmit the Interest message sooner when they have low $t_\alpha$ (Section 4.2.6). When a node has low $t_\alpha$, the Interest is sent faster to the content source (in case of Interest retransmission) and, thus, the content source responds with the Data message back faster than when the node has high $t_\alpha$.

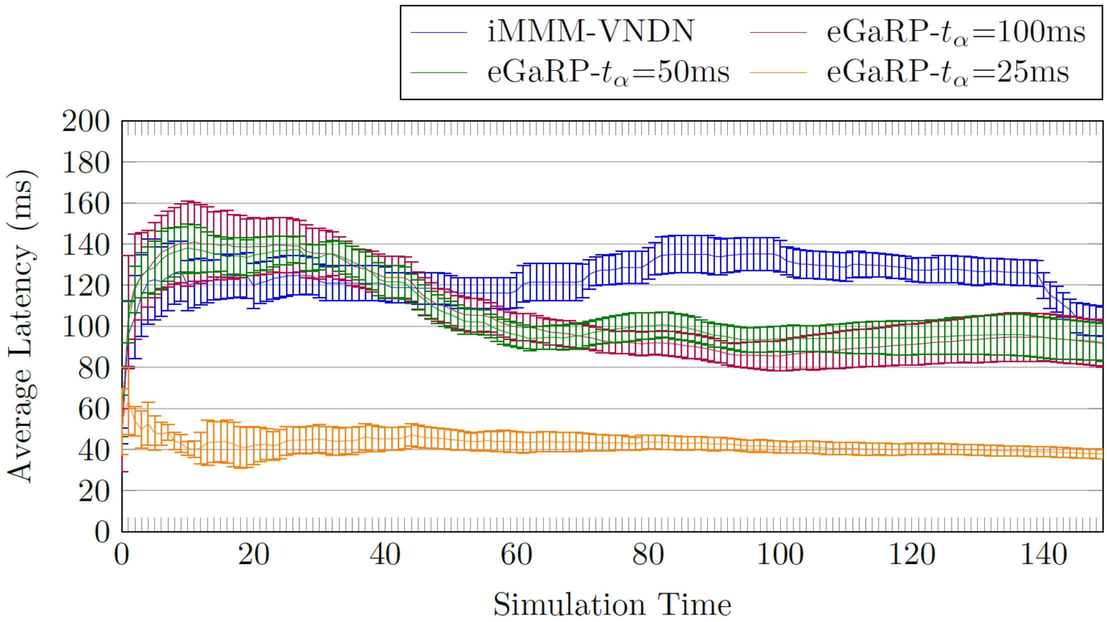



Figure 4.8: Average Latency of the delivered Data in ms in the application layer of the requester node.

Because we use a timer-contention-based forwarding mechanism in each node, we

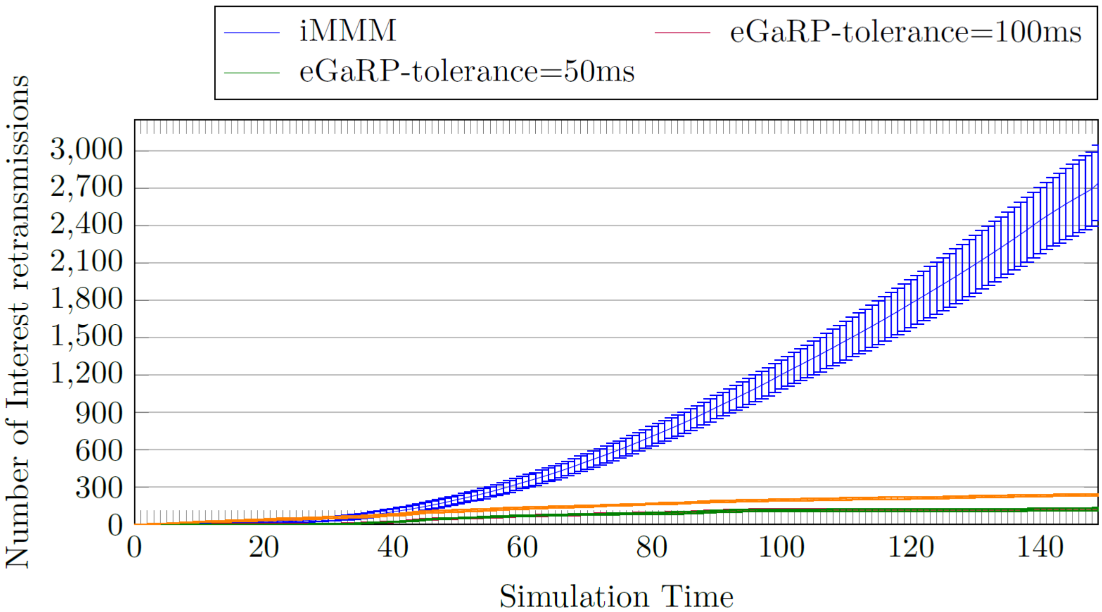



Figure 4.9: Number of Interest retransmissions in the application layer of the requester node.

significantly reduce the Interest retransmissions in the application layer (Fig. 4.9). With eGaRP, each node is responsible for retrieving Data via unicast. Hence, the intermediate nodes are now also responsible for retrieving Data via unicast paths to the requester node. Results depicted in Figs. 4.7–4.9 affect the user's Quality of Experience. The application responds quicker in eGaRP than in [84], reducing the user's waiting time for the content object to load. In addition, the application is responsive, meaning that when it sends an Interest, then the Data message comes back. It does not need to retransmit many Interest messages, affecting possibly the user's experience when, for instance, a user downloads a website. In [84] a user does not retrieve the content object in the 150 seconds that the application runs, meaning that the experience can be interruptive.

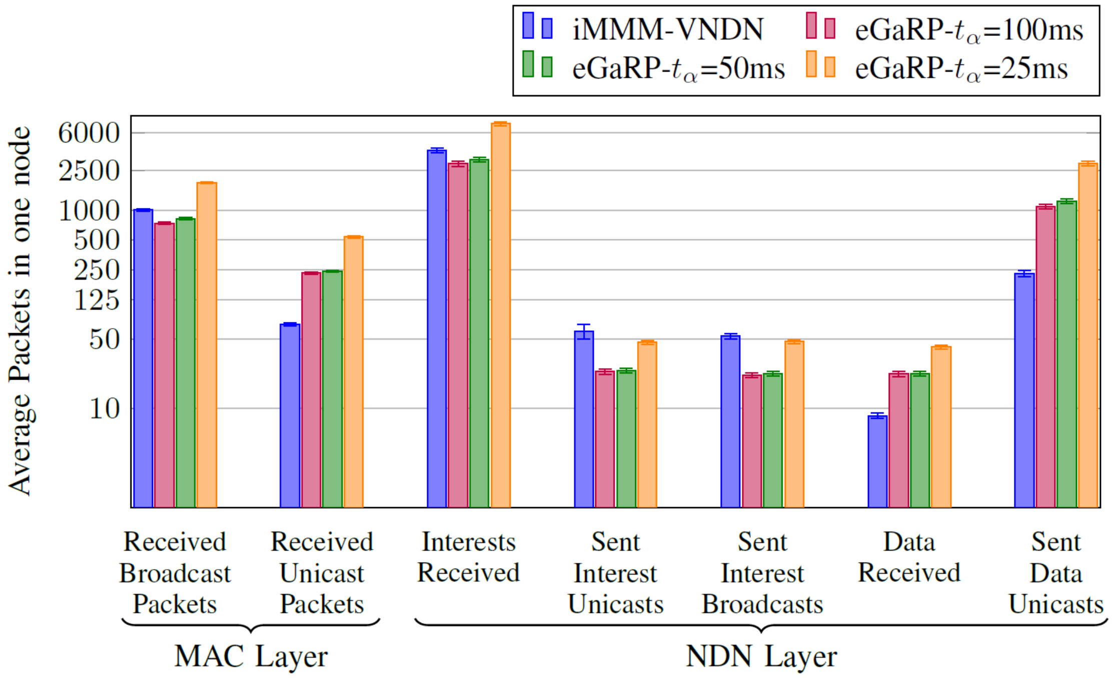



Figure 4.10: Packets in one node in the MAC and NDN layer.

In Fig. 4.10 the other metrics in the lower layers of a node are shown. Fig. 4.10 presents the average number of packets that exist in the network for the total of the simulation time in only one node. The numbers that are shown in the MAC Layer are for 1 out of the 4 interfaces. For eGaRP, the number of messages that are received in a node's MAC layer depends on the $t_\alpha$. When we define a low $t_\alpha$, a node can retransmit more Interests more often, therefore, more messages arrive in a node. After a packet has been received in the MAC layer it is forwarded to the NDN layer. The latter is depicted in the second bracelet of Fig. 4.10. In this layer differences in eGaRP for different $t_\alpha$

and also differences between GaRP and iMMM-VNDN [84] are observed.

1 second is the maximum time a node waits to retrieve a Data message, i.e. the Interest lifetime $t_{INTlif} = 1s$. A node with low $t_\alpha$ receives more messages (both Interests and Data). When the tolerance of a node is small, i.e. $t_{Tol} = 25h\,ms$, its PIT entries will expire sooner than when $t_{Tol} = 100h\,ms$, considering that $h$ is the same in both cases. A node, in that case, can retransmit the same Interest, until the Interest expires, up to $t_{INTlif}/100h = 10h$ times (when having tolerance $t_{Tol} = 100h\,ms$). Similarly, a node can retransmit an Interest up to $t_{INTlif}/25h = 40h$ times, when having tolerance $t_{Tol} = 25h\,ms$. Therefore, in the same time period, when the tolerance is low, nodes can retransmit the same Interest message more frequently, compared to when the tolerance is high. Hence, when $t_\alpha$ is low, there may be more retransmissions of messages from nodes indicating more messages existing in the network, leading to more received Interests and Data in the MAC and NDN layer of a node.

For iMMM-VNDN [84] we see that a node receives more Interests in its NDN layer than in eGaRP for high $t_\alpha$. This is highly correlated with the number of unicast and broadcast Interests a node sends. When a node sends more Interests, other nodes will receive more Interests. An iMMM-VNDN [84] node broadcasts in average more Interests. Thus, other nodes receive more Interests, i.e. more messages arrive at the MAC layer and consequently to the NDN layer. The results presented indicate that for a content retrieval process we manage to unload traffic from a node when using a high $t_\alpha$ by satisfying the application requirements. In contrast, when setting the $t_\alpha$ to a low value, a node experiences more traffic, by again keeping its application requirements satisfied.

## 4.4 Conclusions

The need to design efficient V2V protocols that do not require infrastructure is essential to reduce infrastructure overload and to offload traffic from it. Therefore, in this Chapter, we focus on V2V communication and manage directional forwarding of messages in local areas to support content retrieval. We propose to use directional antennas in vehicles to restrict the geographical dissemination area of a message and, therefore, reduce the amount of load in a node in terms of messages. We also propose a timer-contention-based approach to support the mobility of vehicles in case the path between a content source and a requester vehicle breaks. In particular, we overcome the path breaks by introducing a timer-based retransmission mechanism in vehicles. Our results indicate that in a realistic highly mobile environment we retrieve all the requested Data by keeping the application latency low. By increasing the frequency of possible Interest retransmissions in intermediate nodes, i.e. by using lower $t_\alpha$, we observe a significant decrease in the average latency, which improves the Quality of Experience in drivers, since the drivers do not experience large delays when the requested content object loads.

The main limitation of this Chapter is that the content requester should have a direct or an indirect connection with a content provider. This means that when a vehicle broadcasts a request message, there should be a path consisting of 1 or more vehicles that lead to the content source. If this path exists, then our solution is a viable solution for content acquisition. But, if this path does not exist, the vehicle should either utilize another network, e.g. 5G, or find another way to obtain the requested content object. This idea will be investigated in Part II, where we will use infrastructure to support content retrieval and investigate if and how we should use this infrastructure to improve network performance by supporting path breaks, limit the broadcast transmissions and reduce the dissemination area of transmitted messages.

# Part II

# Vehicle to Infrastructure Communication

In this part, we present routing solutions using Vehicle to Infrastructure (V2I) communication. Although vehicles should be autonomous to perform their operations, V2V communication has limitations mainly because of the proposed standards that are proposed for vehicular networks. In Wi-Fi environments, a large number of connected vehicles leads to channel collisions when vehicles are transmitting messages simultaneously. Therefore, infrastructure can assist in offloading messages from the channel. Moreover, infrastructure can assist in vehicular connectivity by creating paths and insert necessary forwarding information to routing tables of vehicles. In this part, in Chapter 5 we use infrastructure as a main component of our vehicular network, which assists in the routing decisions of vehicles. We use Road Side Units (RSUs) that are deployed on streets as a network component that can assist in routing of requests. We propose that vehicles send their traffic towards the RSUs when vehicles are not aware of how to forward their traffic to find their requested content. In Chapter 6 we use RSUs together with Software Defined Networking (SDN) as the main network component. SDN assists in performing routing path calculations and in populating the routing tables of vehicles. Moreover, SDN instructs RSUs to change their physical layer characteristics, for the latter to be connected with as many vehicles as possible. Finally, we experiment with different node configurations, by installing multiple antennas in vehicles and use infrastructure to identify which antennas should be used for transmitting a message.

# 5

# Infrastructure-Assisted Communication for NDN-VANETs

## 5.1 Introduction

In Chapters 3 and 4 we investigated how we can use V2V communication with NDN in VANETs for content retrieval by developing different algorithms for V2V communication. However, the limitations of V2V communication depend on the limitations of the proposed standards of communication, namely IEEE 802.11p. Therefore, in this Chapter we address **RQ3**, as described in Section 1.2.3, which formulates the question on whether deployed infrastructure, e.g. Road Side Units, combined with ICN and appropriate routing protocols assist content retrieval in VANETs. We use infrastructure, specifically Road Side Units (RSUs), to assist content retrieval in VANETS.

In particular, we propose the use of infrastructure as a possible gateway to connect vehicles in a VANET environment, hence, to improve connectivity, when paths between vehicles break [80]. We use NDN to request particular content that an infotainment application requests in a local area inside a city centre. For routing in NDN-VANETs, the Forwarding Information Base (FIB) tables should be populated and updated when the configured paths are broken. The main advantage of NDN-VANETs is content availability together with data dissemination. In this Chapter, we propose a V2I, and in particular V2R (Vehicle to Roadside Unit), communication architecture for content retrieval in NDN-VANETs: we use infrastructure as a main or a back-up network component that is responsible for the routing of packets. Compared to previous works that typically add additional data structures to the NDN architecture [25, 29, 146], we use:

(a) already proposed data structures by NDN and

(b) RSUs deployed along roads.

We discover content sources, which then advertise their content back to the RSUs. We highlight that the network nodes have no prior knowledge of what content other nodes hold.

Our approach is based on a content discovery phase, which we call *learning phase*. The learning phase creates routing entries in the FIB tables of nodes and the RSU. These routing entries denote nodes that a request can be sent, for the content object to be retrieved. During the learning phase, the RSUs broadcast Beacon messages to discover *content sources*, i.e. nodes that *hold* the content object. The content sources respond to the Beacons by announcing the prefix of their content object. Then, every intermediate node creates two routes: one to the next RSU (from the Beacon transmission) and one to the content source (from the announcement of the prefix of the content source). The learning phase allows:

(i) the RSUs to know the route to the content sources and

(ii) intermediate nodes to create routes to both RSUs and content sources.

After the learning phase, in the *forwarding phase*, a node requests a content object. In the *forwarding phase*, we propose two routing approaches:

- The first one redirects all data traffic requests (i.e. Interest messages) to the infrastructure (in this work to RSUs). Given that the RSUs know how to reach the content sources, the RSUs forward all requests to the content source. Then, the content source sends back the content object to the RSU, and the RSU sends it to the requester.

- The second approach uses the RSUs only as a back-up mechanism: it first looks for a direct route to the content source, if intermediate nodes have routes to the content sources. If no direct route is available, it forwards the Interest message to the RSUs. We highlight that the Beacon broadcast never stops, as the RSUs broadcast Beacons into the network constantly.

In summary, we make the following contributions:

• We propose a V2I and in particular V2R (Vehicle to Roadside Unit) communication architecture that exploits deployed RSU infrastructure for content retrieval in NDN-VANETs.

• Unlike other methods [19, 117], our architecture is resilient to mobility changes, as routes are created and updated in the *learning phase*.

• Our architecture retrieves messages successfully without any prior knowledge of network topology or content availability, thus outperforming previous approaches in terms of content retrieval.

The rest of this Chapter is structured as follows: Section 5.2 describes the architecture of the network and the routing decisions that we made. Section 5.3 shows our evaluation set up and presents a discussion of what we propose in this Chapter. We finally give our conclusions in Section 5.4.

## 5.2 V2R Communication Architecture Description

In VANETs the paths between vehicles are frequently changing, due to the mobility of vehicles. This means that the FIBs should be populated according to newly created vehicle connections. In this Section, we describe the proposed NDN-VANET architecture and analyse the two phases of our V2R communication architecture, the *learning phase* (Section 5.2.1) and the *forwarding phase* (Section 5.2.2). Finally, in Section 5.2.2, we describe the two proposed routing approaches: the linked and the hybrid approach. The main difference of these approaches is how the Interest is propagated in the network. In the linked approach, all nodes send their Interests to the RSU, and the RSU routes the Interests to the content source. In the hybrid approach, nodes send their Interests to the RSU, only when nodes do not have an entry in their FIBs towards the content source.

### 5.2.1 Learning Phase

Our architecture consists of vehicles and RSUs. In the *learning phase*, the routers in vehicles and RSUs are initialized. We treat all routers in RSUs and vehicles as NDN devices that support the NDN architecture, e.g. OBUs that are already deployed in modern vehicles. This initialization is achieved by populating the FIB tables to support the transmission of Interests to the content source. We develop two processes that are used in this learning phase and run endlessly. These processes assist in populating the FIBs of the vehicles and the RSU(s). We summarize the *learning phase* in Algorithm 3. In the following, we describe our two approaches with more details.

**Beacon Transmission.** The first proposed process is called *B-Tr* (Beacon Transmission) and is directly installed in the RSUs. The learning phase starts when the RSUs start running B-Tr, i.e. broadcast a Beacon message. Specifically, the RSU sends a Beacon message in the form of an Interest message. The Beacon has a unique prefix, i.e. "beacon", to be distinguished from other Interest messages (which are requests for a content object), and includes the MAC address of the node (in this case the RSU). We broadcast this message and every node that receives such a Beacon message performs the following steps:

1. It inserts an entry in its PIT and its FIB with the name of the Interest and the MAC address of the node that transmitted the message.
2. It continues broadcasting this Beacon message to other nodes. Every node that receives a Beacon message creates an entry in its PIT and FIB and sends the message into the network.

The created PIT entry assists following the reverse path of the Beacon message back to the RSU, when the response message to this Beacon message arrives. The created FIB entry from the broadcast Beacon message assists in the configuration of paths from all nodes to the RSU.

**Response to Beacon Message Transmission.** The second process is called Response

**Algorithm 3** Learning Phase

**Input:** $Prefix$: Beacon identifier
$RBM$: Response to Beacon Message from content source. .
$D.Prefix$: Data prefix of the content in the content source
$MAC_i$: Previous Hop MAC address
$MAC_j$: MAC address stored in the PIT

```
procedure ROUTING OF A BEACON
    if Beacon received at RSU then
        Create Beacon (Prefix)
        Broadcast Beacon (Prefix)
    else if Beacon received at Intermediate Node then
        Create PITEntry (Prefix, MAC_i)
        Create FIBEntry (Prefix, MAC_i)
        Broadcast Beacon (Prefix)
    else if Beacon received at content source (D.Prefix) then
        Create PITEntry (Prefix, MAC_i)
        Create FIBEntry (Prefix, MAC_i)
        Create RBM (D.Prefix)
    end if
end procedure

procedure ROUTING OF AN RBM
    Receive RBM (D.Prefix)
    if RBM received at Intermediate Node then
        if PITEntry (MAC_j) ≠ ∅ then
            Create FIBEntry (D.Prefix, MAC_i)
            Delete PITEntry (MAC_j)
            Send RBM (D.Prefix) to MAC_j
        end if
    else if RBM received at RSU then
        Create FIBEntry (D.Prefix, MAC_i)
    end if
end procedure
```

to Beacon Message (RBM) transmission *RBM-Tr* and concerns the node that holds the content (content source). When a content source receives a Beacon message, it responds with a new form of a Data message called Response to Beacon Message (RBM). RBMs do not contain any content. The unique name of this message is the name of the content object that the node holds. An RBM follows the reverse path of the received Beacon message. When an intermediate node receives such a message it performs the following tasks:

1. It checks its PIT to find the MAC address to forward the RBM (as in [84]).
2. It creates a FIB entry with the name of the content (as in traditional NDN), and the MAC address of the node that sent the RBM (previous hop), so that an Interest message could reach a content source.

Since many nodes forward the Beacon message, the RBM will also be sent back from the same nodes (because the RBM follows the PIT entries that the Beacon message created). In this way, we populate the FIBs of nodes with many routes that lead to the content source.

The learning phase assists in creating paths from the RSU to the other nodes in the network as well as from the content source to RSU and to other nodes that could possibly request content. Algorithm 3 shows the procedure of processing a Beacon and an RBM message in the learning phase of our architecture. In particular, in Algorithm 3 lines 1-14 we present the Beacon routing process. If a Beacon has been received in an RSU, then the RSU will broadcast the Beacon with its Prefix, i.e. /RSU. If a Beacon has been received in an intermediate node, i.e. to a node that does not request or have the content, then the intermediate node creates a PIT and a FIB entry and continues broadcasting the Beacon. Finally, if the Beacon has been received by a content source, the content source creates a PIT and a FIB entry and creates the RBM message to send it towards the RSU. In Algorithm 3 the next process described is the RBM routing in lines 15-26. If the RBM has been received in an intermediate node, the node checks its PIT to identify if the node received a Beacon message. If yes, then the node creates a FIB entry, deletes the corresponding PIT entry and forwards the RBM. When the RBM has been received at the RSU, the RSU creates a FIB entry and stops the transmission of the RBM.

**Example.** We present an example of the learning phase in Fig. 5.1. The RSU initiates the *learning phase* by broadcasting a Beacon message with a prefix "RSU" and its MAC address "01". Every node in its vicinity will receive the message, create a FIB entry and forward the message. For instance, node A receives the Beacon from RSU and creates a FIB entry with: "/RSU, 01". Node B receives the message from node A and creates a FIB entry: "/RSU, 02". This process continues until the Beacon arrives at the content source. When the content source receives the Beacon message, it responds with an RBM that contains only the content prefix and its MAC address and sends it back to node B (Fig. 5.2). For instance, in Fig. 5.2 the RBM from the content source to node B

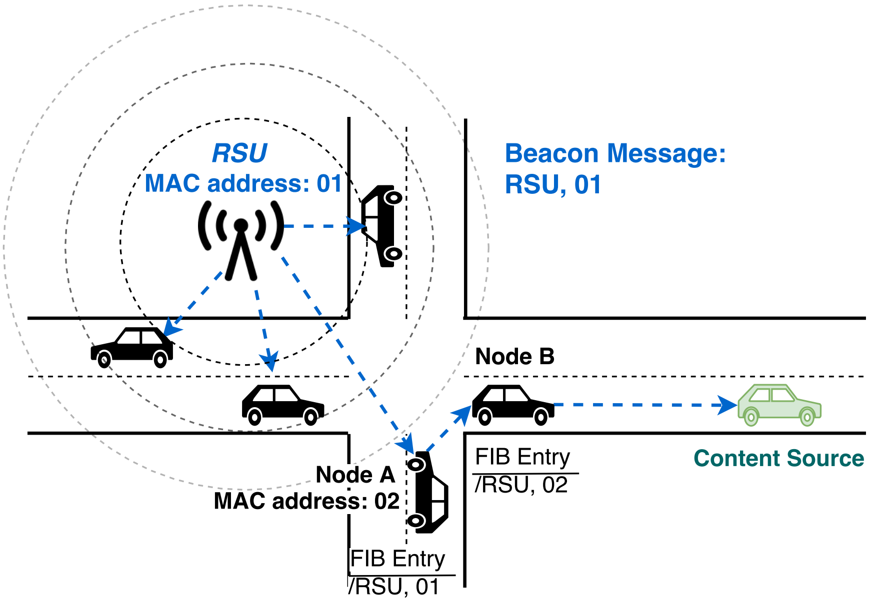



Figure 5.1: FIB tables in *learning phase*. The RSU broadcasts a Beacon message. The message is propagated to all nodes until it reaches the content source.

contains "/video, 04". Then, node B receives the RBM and creates a FIB entry with: "prefix, MAC address", i.e. "/video, 04" (Fig. 5.2). The process continues until the RSU receives the RBM, i.e. until node A forwards the RBM to the RSU, as shown in Fig. 5.2.

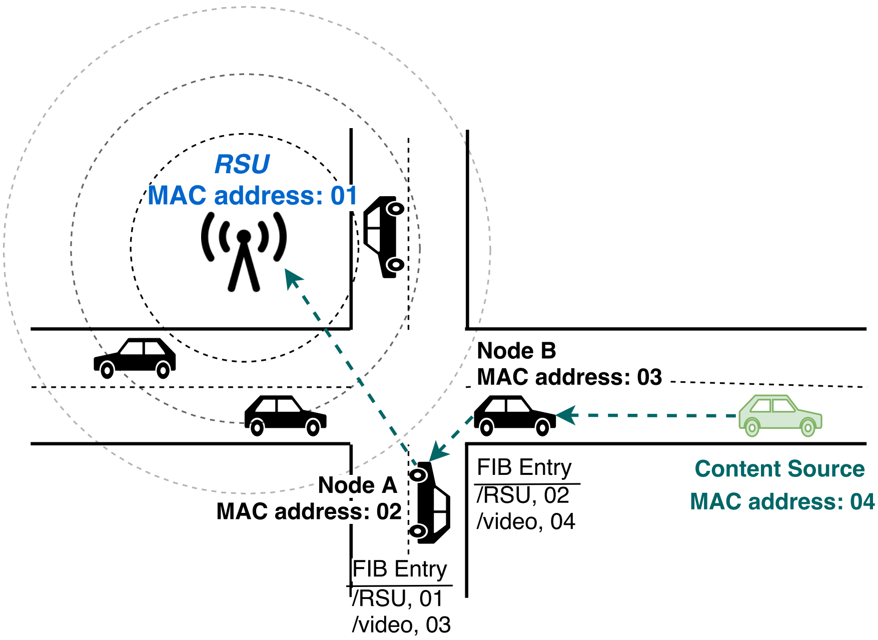



Figure 5.2: Node's A and B FIBs after *learning phase*.

Table 5.1: FIB OF NODES AFTER LEARNING PHASE

| Prefix | MAC address |
|---|---|
| RSU Beacon prefix | 00:01 |
| Content name | 00:03 |

The FIB of nodes resulting from the *learning phase* (Section 5.2.1) is shown in Table 5.1. There is no guarantee that the FIB will contain both of these entries, but the processes described above will always run, so we can assume that after some time the FIBs of the nodes will contain both entries.

### 5.2.2 Forwarding Phase

Since the FIBs are populated, a requester initializes the *forwarding phase*, by routing a content request to a specific source. In our architecture, there are two possible destinations for the Interest message: the RSU and the content source (Table 5.1). We select the next node to route the Interest message according to these FIB entries. This leads to the development of two different routing approaches, according to the selection of the FIB entry: (a) *linked* approach, where a requester always routes the

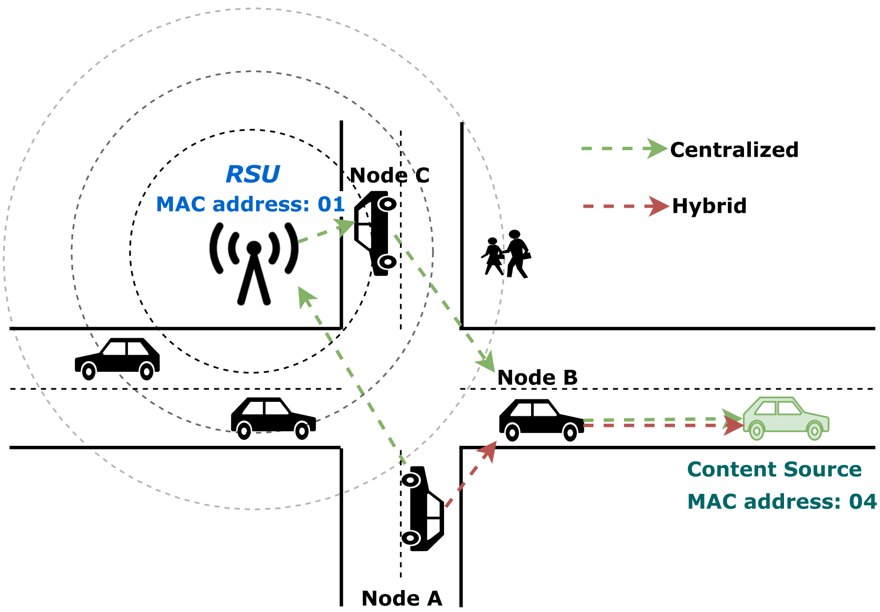



Figure 5.3: Linked and Hybrid paths.

Interest messages along the path to the RSU, and (b) *hybrid* approach, where we exploit V2V and V2R communication by propagating messages using the path to the content source or to the RSU if no route to the content source is available. Fig. 5.3 illustrates an example of the two approaches.

#### Linked Approach

In the linked approach, a node that requests content always routes the Interest messages along the path to the RSU. In particular, it chooses the corresponding FIB entry and sends the Interest to the RSU according to the selected next hop, including the MAC address of the FIB. If no such entry exists, the node waits until a FIB entry to the RSU is created. The transmission processes of the *learning phase* never stop, assuring that a FIB entry is created during the request process (Section 5.2.1).

In case of multiple FIB entries, the requester and all intermediate forwarders choose the entry that is most recently created. This ensures that the selected next node to transmit the Interest message is in the vicinity of the previous node that forwarded the Interest, thus avoiding out of range transmissions. When the RSU receives an Interest message, it searches its FIB to identify routes to the content source. It chooses the most recently created entry (i.e. MAC address) and forwards the message to the selected node. If multiple MAC addresses are available for the same content, once again the RSU chooses the most recently added FIB entry. Next nodes that assist in the forwarding process select also next hops from the FIB. Selecting them ensures that the message arrives successfully at the content source.

Each node chooses an entry from the FIB for routing an Interest. As there are two types of final destinations (RSUs and content sources), there are two possible entries in the prefix of a FIB entry: "RSU", when the final destination is an RSU, and "content name", when the final destination is a content source. A node needs to know which entry to select. We enforce that every Interest message contains a binary flag element. The value of the flag determines the final destination of the message: "0" for RSUs, and "1" for the content source. Therefore, each node chooses entries from the FIB, whose path leads to the desired final destination. When an Interest message is created, its flag is set to "0". When the flag is set to "0", all nodes send the Interest message to the RSU, i.e. all forwarders choose the FIB entry with the RSU prefix. When the Interest message arrives at the RSU, the latter changes the Interest message's flag to

"1" and routes the message according to the entry that leads to the content source. Then by checking the flag of the Interest message that is set to "1", every intermediate node chooses FIB entries that lead the Interest to the content source.

**Hybrid Approach**

After the *learning phase* as described in Section 5.2.1, the FIB tables of the nodes have been populated. In our proposed *hybrid* approach we combine both iMMM-VNDN (c.f. Section 3) and the linked approach by exploiting both V2V and V2R communication models. A node that requests or forwards content objects selects the FIB entry that corresponds to the next node that is leading to the content source.

As mentioned at the beginning of Section 5.2.2 the FIB might not contain an entry to the content source. This can be due to various reasons: (i) the requester node requests content before the content source responds with an RBM, (ii) the RBM has not yet arrived at the requester, e.g. when the content source is multiple hops away, or the message has timed out, or (iii) the Beacon message has not yet arrived at the content source, e.g. when a new vehicle has stored new content in its CS. In these cases, the node sends the Interest to the entry that leads to the RSU. On its way, if the Interest encounters a node with a FIB entry to the content source, it is routed towards the content source; otherwise, it reaches the RSU. In that way, we exploit simultaneously the V2R and the V2V communication model: in particular, we use the RSU as a backup mechanism for V2V to retrieve content when there is not a FIB entry to the content source. If a FIB entry does not exist (neither to the RSU nor to the content source), the requester node delays its request and waits for the creation of a FIB entry.

Fig. 5.3 shows the *forwarding phase* for both the linked and hybrid approaches. In the linked approach, node A sends the Interest message (with its flag set to "0") to the RSU (1st entry in Table 5.1). Then, the RSU changes the Interest flag to "1", searches the FIB for the next hop (node C) leading to the content source, and forwards this message to it. Afterwards, the intermediate nodes follow the same process, i.e. node C sends the message towards node B, and node B sends it towards the content source. In the hybrid approach, we better exploit V2V communication by searching for routes that lead to the content source. In particular, node A has an entry to the content source through node B (2nd entry in Table 5.1). Therefore, node A sends the Interest towards node B, and after node B sends it towards the content source.

In both approaches, the FIB table is deleted regularly in all nodes, including the RSUs. This is a key advantage of our proposed method for two reasons: First, due to node mobility, if no FIB entries were deleted, then nodes will have invalid entries. Using an invalid entry to send an Interest would occupy the channel with unnecessary transmissions and, hence, it would be a waste of resources. For instance, a node would route an Interest to the next hop that might be not there any more, resulting in the expiration of the Interest and waste of resources. Second, it helps to keep the size of the FIB table manageable. For instance, if the size of the FIB is large, a node should perform computationally expensive techniques to retrieve, edit, and/or remove a particular entry.

## 5.3 Performance Evaluation

### 5.3.1 Simulation Environment

#### Simulation Scenario

We use the map of Manhattan that consists of a 1km x 1km grid. There are two lanes for each street into different directions. Second, we use the map of Luxembourg city, i.e. LuST scenario [47], where we isolated an area of 1km x 1km from Luxembourg city centre and extract the mobility traces in this area for different time periods.

#### Simulation Parameters

In this Chapter, since we use only omnidirectional antennas to RSUs and to the vehicles, to implement our proposed architecture, we used the ns-3 [121] based simulator ndnSIM [18, 19, 106] (c.f.Section 4.3.1). We used the SUMO [36] simulator to generate mobility traces of the vehicles.

For both maps and all strategies, each vehicle is equipped with three different Wi-Fi interfaces to send and receive messages at the same time, using IEEE 802.11a Wi-Fi. We use one requester and one content source to obtain our results. Each vehicle uses the Least Recently Used (LRU) cache replacement technique for its Content Store. The number of vehicles is varying from 20 to 400 for the Manhattan map and from 15 to 392 for the Luxembourg map. We experiment with difference number of vehicles in each scenario to test the scalability of our approach. In both maps, we place one RSU as an additional node in the middle of the map (500m, 500m). The RSU is also equipped with three Wi-Fi interfaces, same as in vehicles. A Beacon message is generated by the RSU (almost) every second. We added a random variable to ensure that the Beacon generation will not be the same in every simulation. The requester node sends 10 Interests per second. The learning phase starts from the second 0 of the simulation to instantiate our FIBs. The forwarding phase (requests) starts at second 20. We allow 20 seconds for the learning phase assuming that this is enough time for the FIB tables to be initialized. Our simulation runs for 145 seconds for the Manhattan scenario and 200 seconds for the Luxembourg scenario. The presented results have a confidence interval of 95%. In our experiments, we delete the FIB table every 10 seconds.

In all simulations, each Interest message requests a different fragment of the content, so the cached content should not affect our results. We aim to show how our proposed strategies should behave when an initial request is being issued by a requester. We compare our results with other NDN routing strategies. In particular, we compare our approaches with the flooding approach, where each node broadcasts the Interest message and the Data message is broadcast back following the PIT entries of the Interest [19], with our previous V2V algorithm, iMMM-VNDN [84] and with the AODV routing protocol [117]. We allow AODV 20 seconds to configure the nodes' routing tables, to be comparable with our learning phase. We also used three different network interfaces in the nodes and the interval of the HELLO messages is 1 second.

#### Simulation Metrics

To evaluate our proposed algorithm we used 3 metrics:

- *Interest Satisfaction Ratio (ISR)* (c.f. Section 3.3.1). For AODV we define the Packet Satisfaction Ratio (PSR) as the number of packets that have been received by the source divided by the number of Data that the source has sent.
- *Latency* (c.f. Section 4.3.1). We use the first time that an Interest is sent and do not update the time when an Interest retransmission occurs.
- *Delivered Data* (c.f. Section 4.3.1). For AODV we measure the number of delivered packets, without considering any other message, i.e. RREQ, RREP and RERR.

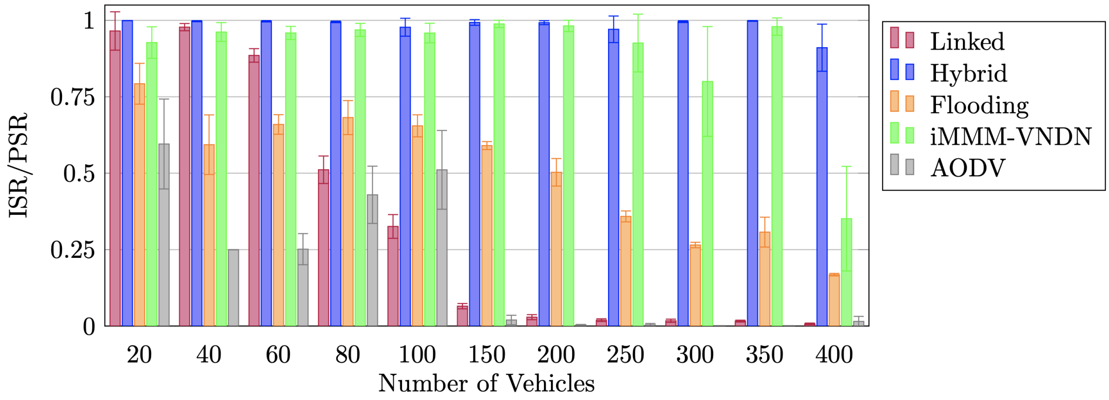



Figure 5.4: Interest/Packet Satisfaction Ratio as function of vehicles for the Manhattan map.

The above metrics describe the main characteristics that we consider important in a VANET for an infotainment application. ISR is a core metric, but it is insufficient, when a VANET application has delay constraints. Hence, latency is measured in our experiments. Moreover, in the ISR, if the number of retransmitted Interests is not measured, i.e. if we only show the ratio of received Data messages at the requester node to the number of different Interest issued, then we could not know how many Data messages the requester node receives. Therefore, we also show how many Data packets are delivered to the requester node.

### 5.3.2 Simulation Results

#### Manhattan Map

Fig. 5.4 presents the ISR/PSR for all strategies. By using the hybrid routing strategy we see that RSU support clearly provides stability in the network since the ISR (for the hybrid approach) is higher than for the others. ISR is higher than 95% for the hybrid approach and for iMMM-VNDN when the number of nodes is low, but for many nodes, we observe that the hybrid approach outperforms iMMM-VNDN in terms of content retrieval. The linked approach has a high ISR for a low number of vehicles, but as the number of vehicles increases, the ISR decreases. In Fig. 5.4 we notice that with AODV the PSR is low. The main reason for the PSR being lower for AODV than the ISR for the hybrid routing strategy is that when a path breaks in AODV, the network wastes a lot of resources to reconfigure a new path by sending control and error packets. These packets increase the congestion in the network and, thus, new routes are not used

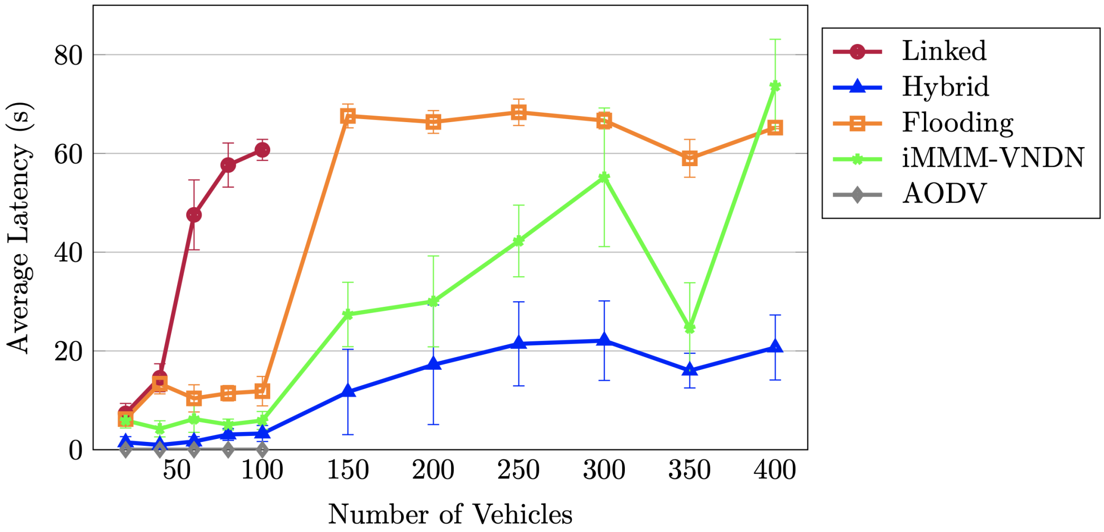



Figure 5.5: Average Latency as function of vehicles for the Manhattan map.

properly. In contrast, in the hybrid approach, we delete the routing information (FIB entries) every 10 seconds (Section 5.2.2) to discover new routes or to update existing ones. In addition, we always choose the route that has been added last to the FIB to ensure that the path will be valid.

In Fig. 5.5 we notice that in the linked approach, the average delay is up to five times longer than for the hybrid routing strategy for a low number of nodes. Redirecting all Interests to the RSU creates a bottleneck around it. Thus, the RSU will reject any message until the congestion around it is resolved. For this reason, the latency is high since rejected Interests need to be retransmitted. For a more dense network, we see that the latency for the linked approach is not illustrated, since almost no messages are delivered to the destination, hence the latency cannot be calculated. We also observe that the hybrid routing strategy results in low latency for a low number of nodes, and as this number increases, so does the latency. This is because we broadcast a Beacon message in the learning phase, and the more nodes exist in the network the more nodes will rebroadcast the message. The RBM will need to pass through more vehicles to create routes. When the number of nodes is high in the network the routes are more vulnerable to break, since a node that moves away will cause several path breaks. We also observe that in iMMM-VNDN the latency increases as the number of nodes increases. This delay is due to the requester node that broadcasts an Interest message every 10 seconds. When more nodes have to broadcast this Interest, the traffic increases and congestion is caused, something that leads to high delay. In

Fig. 5.5 we observe for iMMM-VNDN a decrease in the average latency for 350 vehicles. This happens because the requester node is only a few hops away from the content source and the path does not break, thus the latency of the path is small.

In Fig. 5.5 we see that the latency produced by AODV is much smaller than the average latency of all other approaches for a sparse network. This is because in AODV messages are being sent at the beginning of the simulation when paths have been established after the routing table has been configured with paths. But as the simulation time progresses, paths break and, thus, become invalid. As seen in Fig. 5.4 packets are not delivered any more and the PSR for AODV is less than 1% for many nodes. For a network with more than 100 nodes, no message is delivered to the requester and, hence, the latency cannot be calculated and it is not illustrated. In the hybrid approach, we see that we achieve much higher delivery ratios at the cost of increased latency, caused by the increased number of Interests that exist in the network as the simulation time continues.

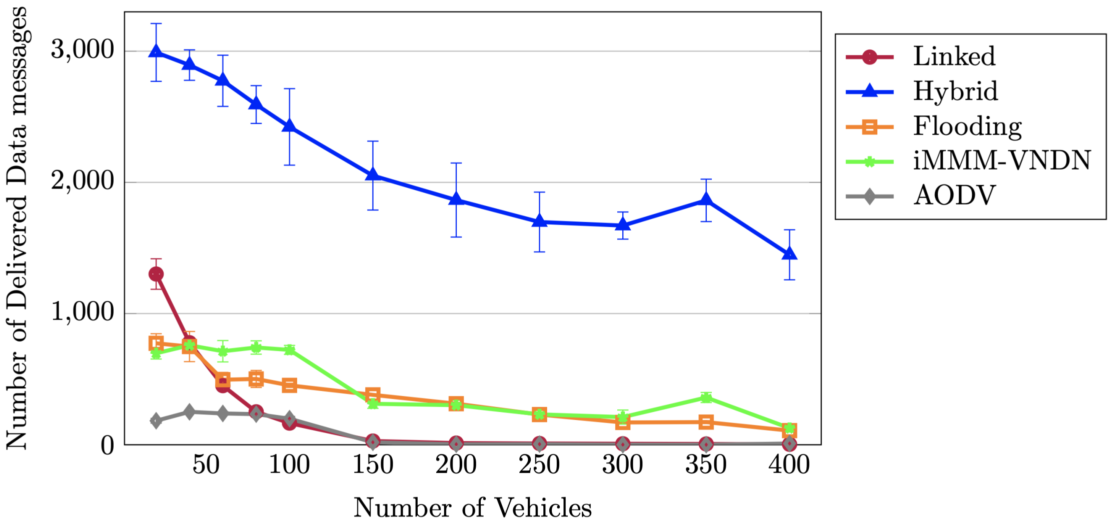



Figure 5.6: Number of Data delivered as a function of vehicles for the Manhattan map.

In Fig. 5.6 we see the number of Data messages that arrived at the requester node. In the linked approach for a low number of nodes and in the hybrid approach for all node densities we manage to deliver more Data messages to the requester. This is achieved because we exploit the number of interfaces that are installed in the requester node (three interfaces using the same frequency) for both approaches. We developed a process installed in the requester node that sends at the same time as many Interests as there are network interfaces installed in a node. In this case, every time an Interest has been issued from one interface, the other two interfaces are also issuing Interests (to be sent to the network). So, we send three Interest messages into the network at the same time. We observe from Fig. 5.6 that the linked approach delivers more Data for a low number of nodes, and while this number increases the number of Data messages delivered decreases. This is because a bottleneck is caused around the RSU and the ISR drops off for many nodes, as seen in Fig. 5.4. The hybrid routing strategy outperforms all other strategies and sends five times more Interests than the flooding approach, three times more Interests than iMMM-VNDN, and ten times more Interests than AODV. In AODV the number of requested packets is low and almost 0 for many nodes because the network is congested. The congestion in the network is caused by the exchange of the control and error messages as well as the high processing load at each node. In addition, all the control and error messages that AODV produces require a portion of the network's bandwidth and, thus, fewer packets from the requester can be sent through the same channel. Therefore, fewer packets are issued by the requester and less Data messages arrive back to the requester node.

The main advantage of the hybrid routing strategy when it is compared to the other strategies is high throughput. The hybrid approach delivers more Data to the requester node (Fig. 5.6) by maintaining a very high ISR (Fig. 5.4). Below, we compare the hybrid approach with iMMM-VNDN. Let us consider that the average bandwidth is $BW$, and the simulation time is the same $T_S$. Moreover, let us assume that the number of received data is $D$. $D$ is different when the number of nodes is different.

The number of Delivered Data for both of the approaches is shown in Fig. 5.6. From Fig. 5.6 we may see that:

$$D_{HYBRID} \simeq 3D_{iMMM} \tag{5.1}$$

where $D_{HYBRID}$ is the number of received data of the hybrid approach and $D_{iMMM}$ is the number of received data of iMMM-VNDN. Since $BW_{HYBRID} = BW_{iMMM}$, $T_S$ is the same for both algorithms and Throughput can be simply defined as $Throughput = {}^{D}/_{T_S}$. Then, it results from (1):

$$Throughput_{HYBRID} \simeq 3Throughput_{iMMM} \tag{5.2}$$

where $Throughput_{HYBRID}$ is the throughput of the hybrid approach and $Throughput_{iMMM}$ is the throughput of iMMM-VNDN. Thus, the throughput of the hybrid approach is almost three times higher than the throughput of iMMM-VNDN.

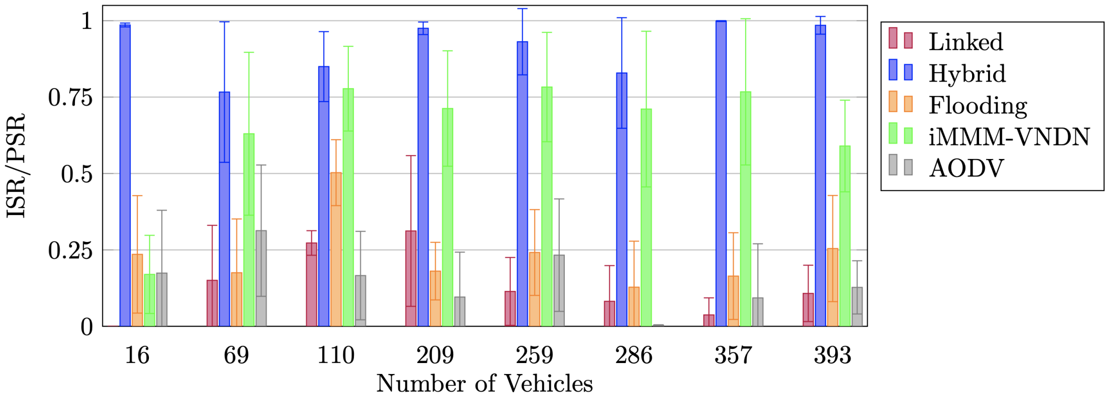



Figure 5.7: Interest/Packet Satisfaction Ratio as function of vehicles for the Luxembourg map.

**Luxembourg Map**

In this simulation, we used mobility traces from the Luxembourg city centre [47]. We isolated an area and use different time slots to extract the mobility in this particular area. We observe that depending on the time, the number of cars travelling through this area changes. In this scenario, we also have one requester node and one node that provides the content. Fig. 5.7 shows the ISR and the PSR for all tested strategies. As in the Manhattan map, the hybrid approach outperforms all others since ISR is higher than 85% for all network densities. We see that iMMM-VNDN has the second best performance in terms of ISR followed by AODV and flooding, depending on the density of the network.

Fig. 5.8 presents the average latency for the strategies. The linked approach has the highest latency for a small number of nodes. For a large number of nodes, we observe that the linked approach has low ISR and the number of Delivered Data is almost zero. This is due to the fact that all Interest messages need to pass through the RSU, to be routed to the node that holds the content. Moreover, we see that for 209 vehicles, the average latency spikes for the linked approach. When 209 vehicles exist, the ISR for the linked approach is the highest. This indicates that the RSU assists in the content retrieval process, but this leads to increased delay. On the contrary, AODV has the lowest latency, as in the Manhattan map. AODV performs best when the requester and the content source are one hop away from each other. But in different cases, when

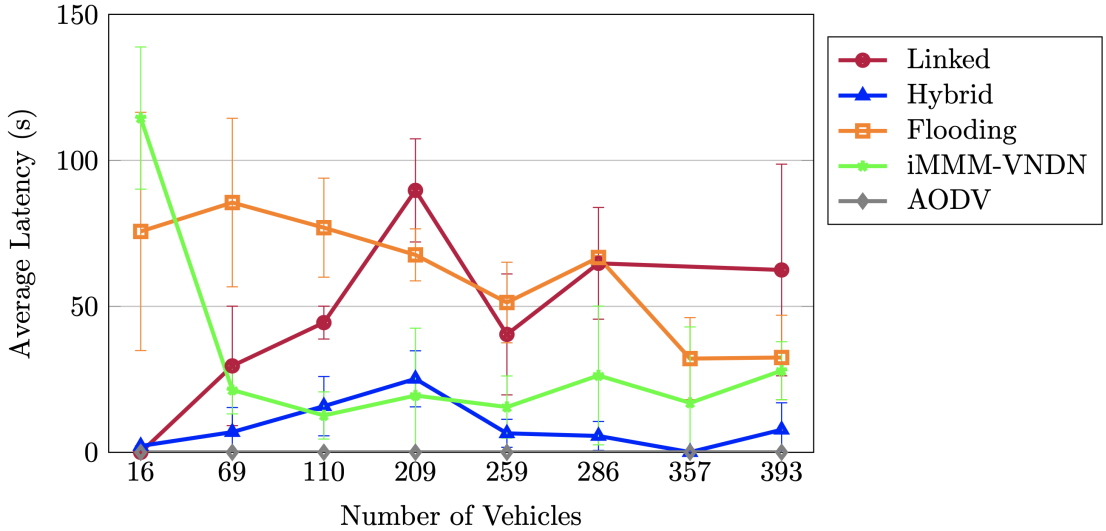



Figure 5.8: Average Latency as function of vehicles for the Luxembourg map.

the two nodes are multiple hops away, AODV fails to receive any messages, since the established AODV paths have changed and, thus, the latency cannot be calculated. The hybrid forwarding strategy has the second smallest delay for almost all network densities. iMMM-VNDN outperforms the flooding and the linked routing strategy in terms of latency but has a higher latency than the hybrid approach and AODV.

Fig. 5.9 shows the number of Delivered Data messages to the requester node. Clearly,

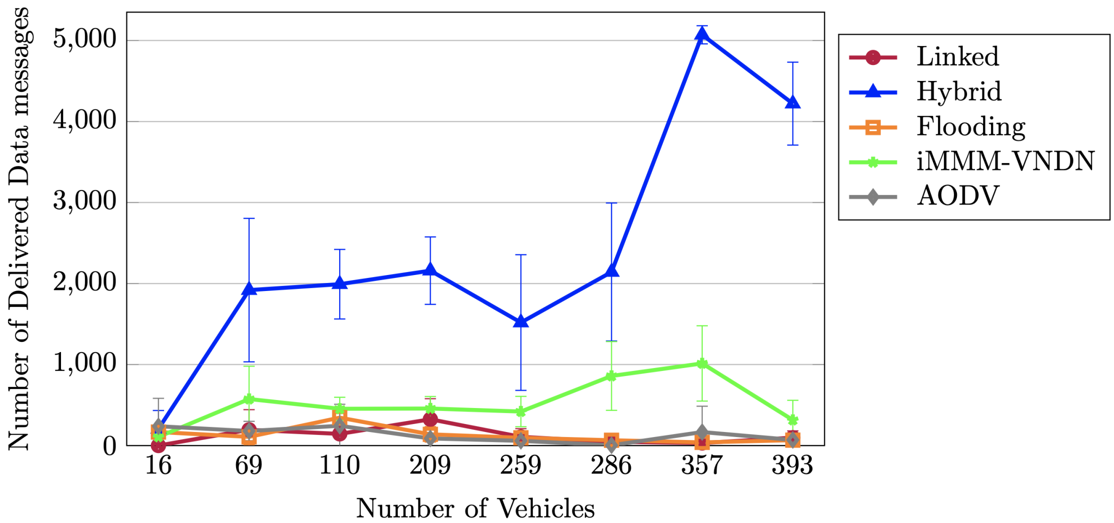



Figure 5.9: Number of Data delivered as a function of vehicles for the Luxembourg map.

the hybrid approach delivers more messages than the other four approaches. The linked approach, flooding, and AODV do not perform well, since they manage to deliver a few Data messages to the requester node, sometimes, less than 100. iMMM-VNDN, on the other hand, performs better than the aforementioned strategies, since, due to the limited broadcast of Interests, iMMM-VNDN is almost resilient to mobility changes. Still, the hybrid approach performs four times better than iMMM-VNDN. As mentioned in the Manhattan scenario, the number of delivered Data is higher for the hybrid approach, since we developed an application for the requester node that sends at the same time as many messages as the number of interfaces that are installed in the requester node, i.e. three messages at the same time. Thus, requesting more Data leads to retrieving more Data to the requester node.

In our presented scenarios, for both routing strategies, i.e. hybrid and linked approach, we start the content request from the requester after the FIBs have been populated. If a vehicle requests a content object without having any entry in the FIB, it waits until a Beacon message or an RBM creates such an entry. For our simulation results, we used a window of $w$=20 seconds for the *learning phase*. We used 20 seconds, assuming that this is enough time for the FIB tables to be initialized. We noticed that, for instance, in the Manhattan scenario both entries are created in the FIB of our requester node after $t$=17 seconds of simulation, see Table 5.1. To examine the range of time $t$ required for both FIB entries to be created in the requester, we experimented with various values of the learning phase's simulation window $w$. Our results show that $t$ remains constant and it is independent of $w$, indicating that varying the learning phase's simulation time $w$ does not result in changes in the created time $t$ of the FIB entries. For instance, reducing the learning phase window to $w$=10 seconds results again in $t$=17 seconds. This is because the creation of the FIB entry depends on both the mobility of vehicles and the time that a beacon message will be received in a node. The mobility of vehicles is the same, and we do not change the frequency of beacon transmission. Instead, we only change when the content request starts. Therefore, the $t$ remains unchanged.

## 5.4 Conclusions

In this Chapter, we presented a V2R approach for NDN-VANETs. We created an architecture that consists of vehicles (in each of them an OBU is installed) and RSUs. RSUs act as a gateway for content requesters to obtain the requested content object. In particular, we designed a learning phase for our network, where an RSU is broadcasting a Beacon message that contains its MAC address and the content source responds with a Response to Beacon Message (RBM) that contains its content prefix. In the learning phase paths are configured from the RSU to all nodes and back, as well as from the content source(s) to all nodes. Then, we designed the forwarding phase, where we investigate how we can use infrastructure and the knowledge we obtained from the learning phase to support content retrieval by reducing broadcast transmissions and by creating paths. We developed two approaches for using the infrastructure: the linked and the hybrid approach.

In the linked approach, we send all Interests to the RSU that is responsible for routing all traffic. In the hybrid approach, we use the RSU as a back-up mechanism to route requests when content requesters do not have an active entry in their routing tables. We evaluated our approaches and our results showed that we achieved the best output in terms of Interest Satisfaction Ratio and delivered Data messages with the hybrid routing strategy. This is because, in the linked strategy, we see that the network, and in particular the RSU, is congested and a bottleneck is created around it. Redirecting all traffic to the RSU causes this congestion, and, thus, selecting the RSU as a back-up mechanism only to retrieve a content object, as we do in the hybrid approach, could be considered as a congestion avoidance mechanism for the RSU, and results in better throughput for our network. We also compared our approaches with other protocols, both using NDN and without and we highlighted that by using NDN, by dynamically updating routing entries, and by allowing each vehicle to route messages hop by hop (instead of using predefined paths), we obtain more content objects in less time.

The limitation of the approach presented in this Chapter lies in the fact that both content providers, as well as content requesters, should have a direct and active connection to the RSU. But it is possible that because of obstacles, interference, congestion and collisions the vehicles will not have a direct connection to the infrastructure. Therefore, in the next Chapter, we will use Software Defined Networking (SDN) to change the physical layer characteristics of RSUs for the RSUs to connect to as many vehicles as possible. If an RSU is connected with more cars, the possibility of connecting to a content provider increases. Moreover, we will use SDN to perform routing path calculation and to populate the FIBs of vehicles, when a content object is being requested.

# 6

# Using SDN for FIB Population and Transmission Power Adaptation for NDN-VANETs

## 6.1 Introduction

Chapter 5 shows how we can efficiently use deployed RSUs to create paths between a content requester and a content provider. We used omnidirectional antennas in all nodes, something that leads to a spreading area of a message in all directions. In this Chapter, we address **RQ4**, as described in Section 1.2.4, which formulates the question on whether a centralized architecture combined with the integration of ICN improves network performance, in terms of vehicular connectivity and content retrieval, in high density VANETs. We investigate whether centralizing the VANET, by using Software Defined Networking (SDN) combined with the integration of ICN, can improve the network performance, by increasing the delivered Data in the requester nodes as well as reducing the overall messages existing in the network.

In Chapter 5 we concluded that the connectivity between Road Side Units (RSUs) and a large number of vehicles is disrupted in a vehicular environment. In particular, when an RSU tries to communicate with many vehicles, then collisions around the RSU create huge packet loss leading the RSU to reject all incoming messages. Keeping that in mind, in this Chapter we investigate if SDN can be applied into such an environment to decrease the collisions around RSUs by adapting its transmission power [82] and by assisting in the routing of messages in the VANET when a request is issued by a vehicle. SDN decouples forwarding functions (data plane) from network control (control plane), allowing for the development of robust adaptable forwarding schemes, where network components, such as switches, can be configured remotely [93]. SDN's remote management offers convenient deployment, centralized control, reliability and flexibility [93, 150].

Hence, by changing the transmission power of an RSU we change its radius. We highlight that tuning the RSU to always function to its highest transmission power might not allow the latter to connect with the highest number of cars. This happens because highest transmission power creates more interference into the network, and this creates higher collision probability leading to rejection of messages. SDN is applied to the network. RSUs are used as switches and we deploy an SDN centralized controller for programming all network components.

To perform so, we use an SDN controller application (which is deployed away from city streets) to change characteristics (transmission power) of Road Side Units (RSUs) and to assist in message routing. In general, the scope of SDN is to centralize the network. The role of an SDN controller is to:

- have a global network knowledge to assist (future) applications.
- instruct vehicles about satisfying their requests, i.e. service establishment and message transmissions.
- notify vehicles about emergencies, such as road accidents.

We install in vehicles and RSUs multiple antennas to allow vehicles either to perform simultaneous message transmissions or to increase their range (via beamforming techniques). Vehicles are communicating via the control channel using one omnidirectional antenna with each other and with the deployed RSUs. For content retrieval, we investigate two different antenna configurations.

In the first configuration, named Multiple Interfaces Configuration (*MIC*), a content object is retrieved via directional antennas installed in vehicles and RSUs listening to service channels. By using directional antennas we can improve the coverage area of a vehicle and limit the dissemination area of messages. But, installing multiple directional antennas can increase interference. We also allow the rotation of these directional antennas. The difference between the rotation of the directional antennas between this Chapter and Chapter 4, is that the rotation of antennas in the MIC is performed via rotating their beams, instead of mechanical rotation used in Chapter 4 (c.f. Section 4.2.3).

In the second configuration, named *SU-MIMO*, multiple omnidirectional antennas are installed in vehicles and RSUs and we use them as a Single-User Multiple Input Multiple Output (SU-MIMO) system. A SU-MIMO system improves communication by mainly reducing the bit error rate and increasing the bandwidth of the channel [115]. But, nodes can communicate only with one node at a time.

Furthermore, the SDN controller is connected to the RSUs and collects information about vehicular and network traffic from them. Then, the SDN controller calculates the number of connected cars to an RSU and decides whether to change its transmission power, if it calculates an RSU can connect with a higher number of cars. Finally, the SDN controller having both local and global knowledge of the network topology assists in path calculation, when a content request is being issued, and populates the routing tables of nodes participating in the content object exchange process.

The rest of this Chapter is structured as follows: Section 6.2 describes our proposed system model and Section 6.3 describes the content retrieval process. In Section 6.4 we present our results and in Section 6.5 we draw our conclusions.

## 6.2 System Model

In our proposed system we assume that RSUs are deployed in city streets. These RSUs are connected to an SDN controller, as shown in Fig. 6.1. In this work, we populate the routing tables of vehicles (FIB tables). For this, we choose to utilize an SDN controller. The SDN controller (we will refer to it also as controller) has sufficient computational power and can calculate routing paths, as well as instruct the RSUs to change their transmission ranges. The controller instructs the RSUs to change their transmission power when it calculates that the RSUs can connect with more vehicles than they are already connected.

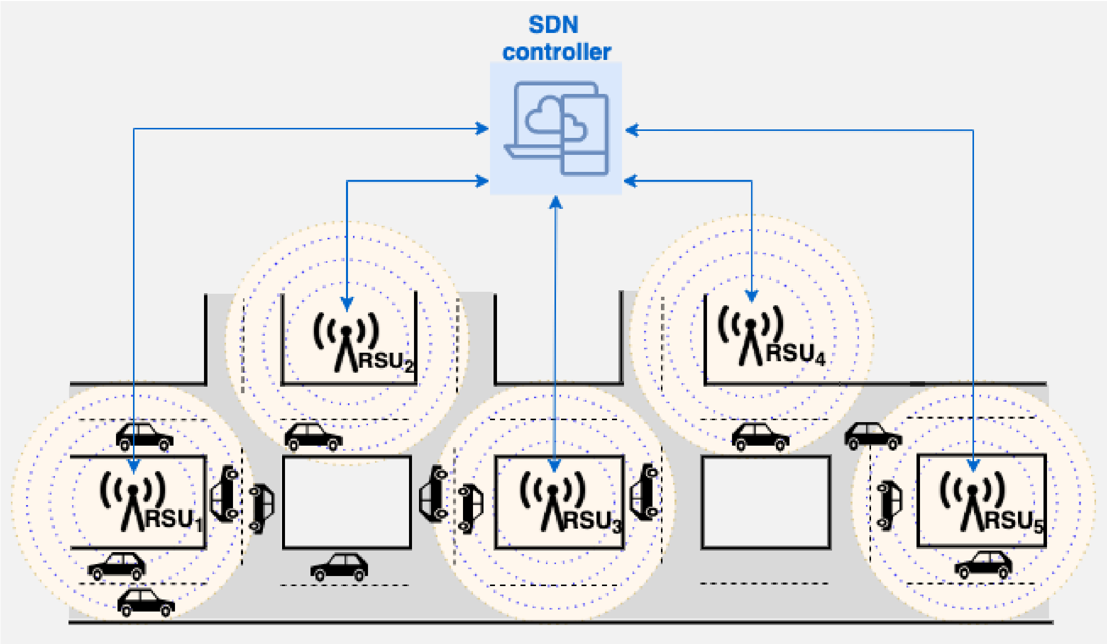



Figure 6.1: Application Scenario

For an SDN controller to identify if it needs to change the transmission range of an RSU, it should know the number of connected vehicles to all RSUs that it controls and the total number of cars (connected or not) inside the coverage area of all RSUs. Therefore, vehicles should have an active connection with an RSU, when they are in its range.

### 6.2.1 Communication Between Network Components

In Chapter 5 every RSU sends periodical messages in its coverage area to discover potential content sources. In this work, we use this periodical message exchange also to identify the vehicular traffic inside an RSU's coverage area. We experiment with two different antenna configurations in nodes (both on vehicles and on the RSUs), as shown in Fig. 6.2.

In the first configuration (Fig. 6.2a) named *MIC* (multiple interfaces), we assume that all vehicles and RSUs are equipped with multiple wireless interfaces, each equipped with a different antenna, running IEEE 802.11p as their communication protocol. We assume that we have $N$ interfaces and $N$ antennas installed in nodes. For these

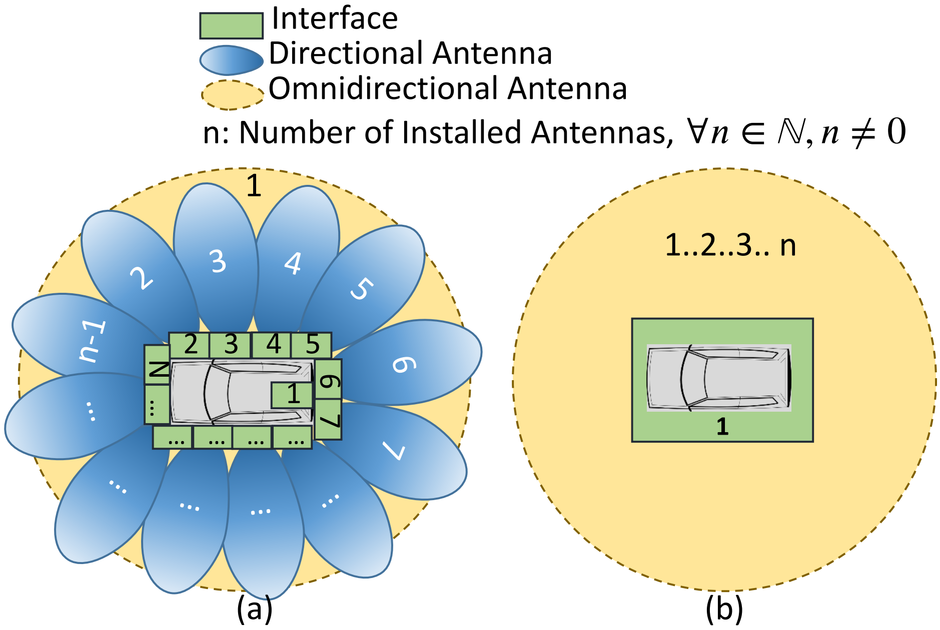



Figure 6.2: Different node configurations. Figure (a) presents the multiple interface configuration (MIC) and Figure (b) presents the SU-MIMO configuration

antennas we choose $N-1$ antennas to be directional antennas and one to be an omnidirectional antenna. Every interface that has a directional antenna attached is listening on a particular service channel of IEEE 802.11p to communicate with other vehicles for content object exchange. The interface attached to the omnidirectional antenna listens to the Control Channel.

In the second configuration, named *SU-MIMO*, we assume that a vehicle has one wireless interface installed, in which $N$ omnidirectional antennas are installed (we adopt the native design of MIMO systems), as shown in Fig. 6.2b. For the one installed interface, we use these $N$ omnidirectional antennas as a SU-MIMO system, to focus their power towards a particular direction. In addition, for both configurations, we install in each RSU an additional interface to communicate with the SDN controller. The communication, therefore, is defined as follows:

- The communication between vehicles for content object exchange via V2V is performed for the *MIC* configuration via their interfaces equipped with directional antennas. For the *SU-MIMO* configuration, this communication is performed via the MIMO system, i.e. via the one installed interface that exists on vehicles by steering its antennas towards the destination vehicle. We make this design choice to reduce the dissemination area of messages when a content object is requested.

- The communication between vehicles and RSU is performed for the *MIC* configuration only with the interface equipped with the omnidirectional antenna and listening permanently to the control channel. As shown in Fig. 6.2a we use only the interface marked with a dashed line.

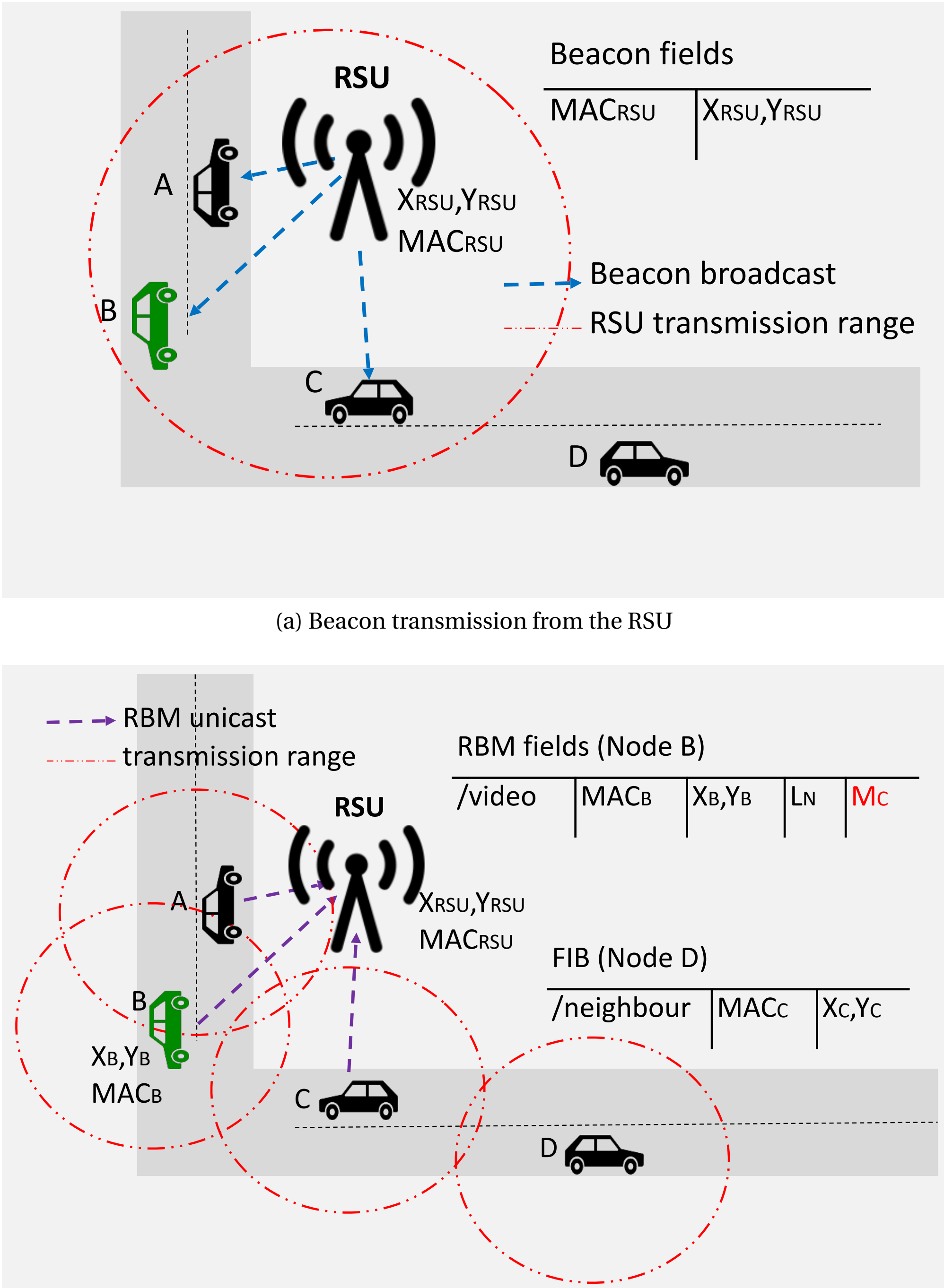
  For the *SU-MIMO* configuration, the communication between vehicles and RSUs is performed via the one interface installed, without performing any actions to configure the installed antennas. As shown in Fig. 6.2b, we only have one interface installed in vehicles and RSUs, and the communication is performed via this interface. For this communication, the interface is also listening permanently to the control channel. For both configurations, we never change channels, since control messages in IEEE 802.11p are exchanged via the control channel.

- The communication between RSUs and the SDN controller is performed via a fixed network.

### 6.2.2 Data Collection by the SDN Controller

The SDN controller should collect all the required data to perform its operations (path calculation and transmission power adaptation). The collection of these data is performed via the RSUs that are deployed on the streets. For this collection, we use for the *MIC* configuration only the interface with one omnidirectional antenna installed and for the *SU-MIMO* configuration the interface without applying any beamforming techniques. Hence, we leave the SU-MIMO system intact.

First, an RSU sends periodic broadcast messages to vehicles. These messages contain the MAC address and the geographical coordinates of the RSU. This is depicted in Fig. 6.3a. As in Chapter 5, when a vehicle receives a message from the RSU it creates a FIB entry containing the MAC address of the RSU, its position and the Signal-to-Interference-plus-Noise Ratio (SINR). The vehicle, then, responds with a unicast message to the RSU, named Response to Beacon Message (RBM), Fig. 6.3b. For the *MIC* configuration, i.e. when a vehicle is equipped with both omnidirectional and directional antennas, the RBM contains the vehicle's identifier, e.g. MAC address, the vehicle's current geographical coordinates and a list of the vehicle's neighbours. The




(a) Beacon transmission from the RSU



(b) RBM response from node B. FIB of node D

Figure 6.3: Data collection from the RSU and, therefore, from the SDN controller

list of neighbours indicates the surrounding vehicles (their MAC addresses and their geographical coordinates) that are connected (1 hop distance) to a vehicle. If a vehicle is a content provider, the message also includes the name of the provided content object. We assume that a vehicle is equipped with a GPS device and, therefore, knows its current geographical position [81]. When the vehicle has multiple omnidirectional antennas (SU-MIMO system) the RBM contains all the fields mentioned before together, i.e. the MAC address, the geographical coordinates and the list of neighbours of the vehicle, with the required information for the *SU-MIMO*, i.e. how the periodic broadcast message that the RSU sent reached all the antennas of the vehicle. This process is shown in Fig. 6.3b. In Fig. 6.3b we assume that node B holds a content object in its Content Store named /video. Node's B RBM contains the name of the content object /video, $MAC_B$, which is node's B MAC address, $X_B$,$Y_B$, which are node's B geographical coordinates and $L_N$, which is the list of node's B neighbours. In this case, the neighbours are node A and node C. For the *SU-MIMO* configuration the RBM contains an additional field $M_C$, which describes how the beacon message arrived at node's B antennas. We can assume that the beacon message serves as the Null Data Packet and the RBM has a field indicating the feedback matrix as in Wi-Fi 802.11 ac [67].

A vehicle responds with an RBM, which is unicast to the RSU via the antenna that points to the RSU. Other vehicles that are around the sender will overhear the RBM transmission. Vehicles overhearing messages that are intended for the RSU, extract the source MAC address and the geographical coordinates of the RBM and insert this information into their FIBs. The name of the created FIB entry is either "/neighbour (indicating that this MAC address is a neighbour) if the vehicle is not a content provider, or the content object name "/video" if a vehicle is a content provider. When a vehicle unicasts a message to the RSU, it checks its FIB for identifying neighbours and inserts this list of MAC addresses and their corresponding coordinates in the RBM message. In Fig. 6.3a node D does not receive a beacon message because it is outside of the coverage area of the RSU. Hence, it does not respond with an RBM, but node C responds with an RBM (Fig. 6.3b). Therefore, since nodes D and C are connected with each other, node D will overhear node's C transmission and enter node's C information ($MAC_C$, $X_C$,$Y_C$) into its FIB as a neighbour node (/neighbour). When the RSU receives an RBM message it forwards it to the controller through the corresponding fixed network interface. Processing of these messages in the controller is described in Section 6.2.3.

### 6.2.3 SDN Controller Functionality

The first function of the SDN controller is to decide whether an RSU should change its transmission range by changing its transmission power to be able to communicate with more cars. As in [132] we can calculate the received power of an RSU considering all losses in vehicular communications:

$$P_r = P_t + G_r + G_t - \sum L_x, \tag{6.1}$$

where $P_r$, $P_t$ are the reception and transmission power, $G_r$ and $G_t$ are the gains of the receiver and the transmitter, and $L_x$ are all the losses of the signal. Losses occur because of obstacles, noise, interference, weather conditions, and thermal noise.

The limitations in the physical layer of the communicating devices depend on the wireless standard that is used. The controller knows that the maximum characteristics of a network device correspond to a certain coverage area.

For the controller to decide whether it needs to change the coverage area of an RSU, it needs to decide whether the number of cars connected to all RSUs can be increased. Therefore, the controller should change the RSUs' transmission power taking into account the total number of cars existing in the RSUs' coverage areas. To change the transmission power, Fig. 6.4 shows how many cars are connected to one RSU in terms of the number of antennas that are installed in the RSU and the vehicles. The results of Figs. 6.4–6.5 are derived from experiments performed in the Luxembourg scenario [48]. The Luxembourg scenario consists of vehicles moving through the Luxembourg city for 24 hours. From the city of Luxembourg, we choose an area of 2km x 2km in the city centre, where we run our algorithm for 300 seconds. The total number of vehicles that pass through the selected area for these 300 seconds is 608. Specific parameters of the selected scenario are shown in Table 6.3. In addition, inside this selected area we install 1 RSU that is connected to the SDN controller and the vehicles. For the experimental results shown in Figs. 6.4–6.5 no content object is requested. Hence, the RSU does not perform any beamforming technique to its antennas and does not use its antennas as a MIMO system. The beamforming technique and the MIMO system are only used in the content retrieval process. So, the presented results are the same for both configurations, since the communication between RSU and vehicles is performed for both using omnidirectional antennas. For the path loss model, we use the two ray model [131, 134], where the gain of the transmitted signal depends also on

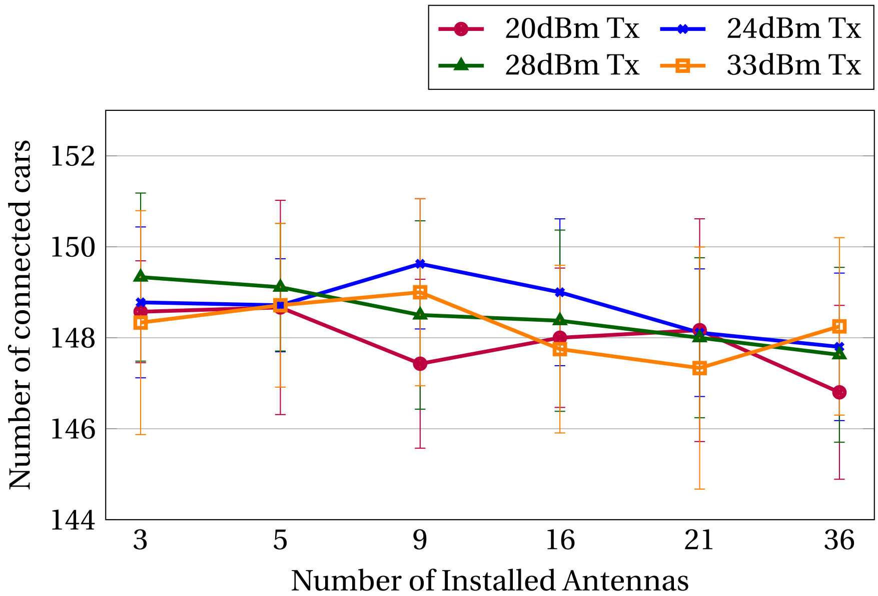



Figure 6.4: Total number of connected cars to the RSU as a function of the number of installed antennas in vehicles and the RSU

the position of the car and the reflection of the signal from roads. Together with the two ray model we use the obstacle path loss model [132, 133], where the signal gain is reduced when it passes through obstacles (such as buildings).

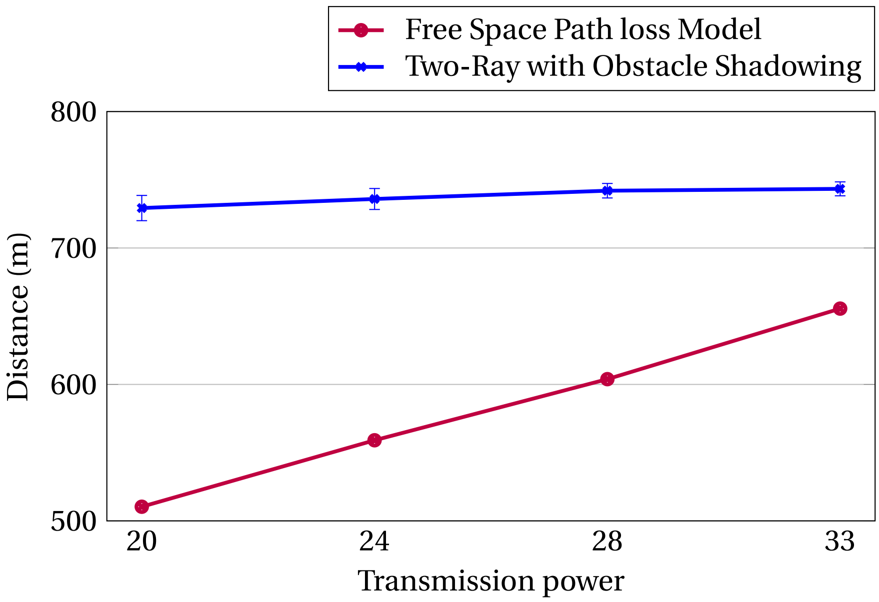



Figure 6.5: The distance of the furthest away connected vehicle from the RSU, when there are 36 antennas installed at the RSU

For the same scenario, we measure the distance between the furthest away vehicle that is connected to the RSU. We present this in Fig. 6.5. Hence, in Fig. 6.5 we indicate approximately the transmission range of the RSU. We compare the simulation results with the free space path loss equation (Equation 6.1). From Figs. 6.4–6.5 we see that when the number of antennas increases we have a minor reduction in the number of cars that the RSU can connect for both configurations. But, we can compensate for this decrease of connected cars by increasing the transmission power of the RSU. The transmission range of an RSU can slightly increase also by increasing the transmission power of the device.

From Fig. 6.4 we can define when the controller should change one RSU's transmission power, based on the number of cars connected to all RSUs at the current time. This means that the controller can change the transmission power of an RSU based on current and previous measurements of the number of cars connected to the RSU. In addition, when the SDN controller is connected with multiple RSUs, it changes the transmission power of each RSU separately, if the total number of connected cars to all RSUs decreases compared to the last time step. In that case, the SDN controller will change the transmission power of each RSU to the value that corresponds to the maximum number of connected cars for each RSU and for the whole network (Algorithm 4). Specifically, Algorithm 4 takes as input the number of connected cars

**Algorithm 4** Transmission Power adaptation

**Input:** $Vc$: number of connected cars to all RSUs
$Vs$: previous number of vehicles connected to all RSUs
$i$ indicates the ith RSU
$Vc_i$: number of connected cars to $RSU_i$
$t_i$: map<Tx,connected Cars> of $RSU_i$
$N_{RSU}$: number of RSUs the SDN controller is connected to
$Tx_i$: transmission power of $RSU_i$
$Tx_{inew}$: new transmission power of $RSU_i$

**if** $Vc < Vs$ **then**
  **for** $i \rightarrow 1 : N_{RSU}$ **do**
    find $Tx_{inew} \neq Tx_i$: $t_i \rightarrow t_i[Tx_{inew}] > Vc_i$
    $Vs = Vs + Vc_i$
    return $Tx_{inew}$
  **end for**
  $t_i$.insert$(Tx_i, Vc_i)$
**end if**

to an RSU at the current timestamp and the number of connected cars to this RSU in the previous timestamp, i.e. how many cars the SDN controller counted that where connected to this RSU. The SDN controller takes this input for every RSU. Then, the controller will change the transmission power of each RSU, if it calculates that the number of cars each RSU is connected to, can be increased. This is performed by mapping each transmission power the RSU has to the number of connected cars in this RSU.

Finally, when the SDN controller receives an RBM it creates a map based on the MAC address of the RBM and on the MAC address of all the neighbour nodes the RBM contains. This map is updated with every newly received RBM and is used for calculating routing paths for content retrieval, as described in Section 6.3.

## 6.3 Content Retrieval

For content retrieval, three possible cases are defined based on the system model, c.f. Section 6.2. A vehicle requesting a particular content object (requester) issues an Interest and checks its FIB to identify a next hop to send the Interest, for the Interest to reach a content provider. Since vehicles exchange periodic messages with an RSU, they have populated their FIBs with entries pointing either to content sources, RSUs and/or neighbour nodes.

1. The first case is when a vehicle has direct communication with a content provider (a node that has the content object), i.e. the vehicle has a FIB entry defined by the content object name. If it has, as in native NDN, the Interest is unicast using the MAC address of the FIB entry. To perform the unicast, first, for the *MIC* configuration, the node sending the Interest will calculate the angle between its own position and the target node based on their geographical coordinates. Then, the requester will choose the interface with a directional antenna that points to the target vehicle. After, the requester will change the beam of this antenna to the calculated angle. Finally, the requester will change the beam of all directional antennas by the same angle, to continue to have a coverage of 360°and will transmit the Interest. For the *SU-MIMO* configuration, the node will calculate the steering angle of its antennas (how to calculate the steering angle and how to enable MIMO in vehicular networks can be found in [22, 42, 111, 115, 147]), will steer its antennas based on this steering angle and will transmit the Interest.

Table 6.1: FIB population (FIBPop) message

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Name | Source | Destination | RAntenna | FIBdest | FIBcoord | VAntenna | SChannel |

2. In the second case, a vehicle does not have direct communication with the content provider, but it has a FIB entry to an RSU. If a vehicle has FIB entries to multiple RSUs, it chooses the entry with the highest SINR. Then, the vehicle unicasts its request to the chosen RSU. The RSU forwards this request to the SDN controller, which:

- searches the created map of vehicular connections to identify possible paths between the content requester and every content source,
- selects one path based on the lowest hop count using the Dijkstra shortest path algorithm,
- creates FIB population (FIBPop) messages to send to the RSU. FIBPop messages are used to populate the FIB (routing) tables of vehicles,
- selects for the *MIC* configuration the interface of the RSU that the FIB message should be transmitted from. For the *SU-MIMO* configuration, the controller calculates the steering angle of the MIMO system of the RSU.

The structure of a FIB population message is shown in Table 6.1. The FIB message contains fields regarding the RSU and the target vehicle. This message contains the following fields:

(a) The name of the content object requested as it should be shown in a node's FIB table.

(b) The Source of the message (which is the RSU).

(c) The Destination of the message that indicates the MAC address of the node that should receive the message.

(d) For the *MIC* configuration, RAntenna (RSU antenna) shows the antenna that the RSU should use to transmit the message. For the *SU-MIMO* configuration, RAntenna indicates the steering matrix that the RSU should use.

(e) FIBdest indicates the MAC address that the node receiving the FIBPop should enter in its FIB.

(f) FIBcoord indicates the coordinates of the FIBdest. This field is intended for the node receiving the FIBPop message and should be entered to the same FIB entry.

(g) For the *MIC* configuration, VAntenna (Vehicle antenna) shows the interface that the vehicle should choose to forward a message. For *SU-MIMO* configuration, VAntenna indicates the steering matrix that the vehicle should use.

(h) SChannel that indicates the selected channel of communication.

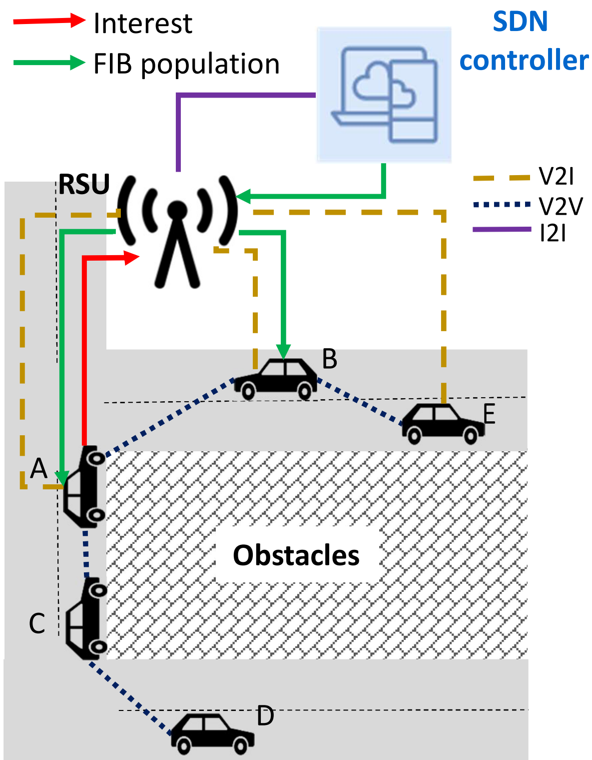



Figure 6.6: Communication and message exchange

Let us assume that as shown in Fig. 6.6 the SDN controller receives an Interest message from node A. The controller checks its map to identify a path between node A and content provider E. It selects a path from node A via node B to node E. Then, the SDN controller sends to the RSU two FIB population messages, as the number of hops that the Interest should go through to reach the content provider (node A and node B). For the *MIC* configuration, the first RBM is sent from a directional antenna $n$ to node A and the second from the directional antenna $j$ to node B. For the *SU-MIMO* configuration, the first RBM is sent to node A using the steering matrix $n$ and the second RBM is sent to node B using the steering matrix $j$. The FIBPop messages are defined in Table 6.2. If node B requests a content object, it will send the Interest to node E directly, since there is V2V communication between node B and node E.

Table 6.2: Examples of 2 FIB Population (FIBPop) messages

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| \video\v1\1 | RSU | node A | $n$ | node B | $x_b, y_b$ | $k$ | sch1 |
| \video\v1\1 | RSU | node B | $j$ | node F | $x_f, y_f$ | $y$ | sch1 |

3. The third case is when there is neither a FIB entry to the content provider nor to an RSU. In this case, the node selects a neighbour node from the FIB table and it unicasts the Interest there. Every neighbour receiving the Interest unicasts it either to a content source, an RSU or a neighbour. This process continues until the Interest reaches a node with a FIB entry either to a content provider or to the RSU. In Fig. 6.6, if node C requests a content object, it will send the Interest either to node A or node D. We note that since the exchange of RBM messages depends on the periodic messages an RSU sends, it is possible that a node will not have a FIB entry. This will happen when a vehicle and all of its neighbours do not have an active connection to an RSU. In that case, we consider that the node can download the content object using another network (e.g. 5G) if the message is urgent (i.e. safety information) or it can wait until there is an entry to its FIB (to a content source, an RSU or a neighbour node). In our case, we choose to wait until a node has an entry to its FIB to unicast the Interest. We highlight that we do not use any broadcast message transmissions for content retrieval. We use broadcast messages for the periodic message exchange from an RSU as it is defined in the IEEE standards as basic safety messages [50]. This avoids overhead in the network, broadcast storms and collisions as well as saves network resources.

## 6.4 Performance Evaluation

This Section describes the evaluated scenarios, the evaluation parameters that were used and the experimental results.

### 6.4.1 Simulation Environment

#### Simulation Scenarios

Our algorithm was evaluated using the Luxembourg city scenario [48], by selecting an area of 2km x 2km in the city centre. Inside this area, we take into account two different RSU placements. In the first, we place 1 RSU to the centre. In the second, we add 5

RSUs distributed inside this area and we connect the SDN controller with all RSUs. For both RSU placements we run our algorithm for 300 seconds during rush hours (6 pm). During the simulation, the controller collects information from vehicles (through the RSU(s)) and assigns new transmission power values to the RSU(s), as described in Section 6.2.3. Random vehicles request a content object for these seconds, and the SDN controller is responsible for creating and choosing routing paths. Vehicles request 150 Interests during the 300 seconds period simulation time, when being inside the selected area. A vehicle may leave the area before sending all 150 Interests. In that case, we assume that there will be another RSU outside of this area for the vehicle to continue the content retrieval process.

We run our algorithm for two different scenarios and for two different RSUs placements (1 RSU in the area and 5 RSUs inside the area), defined by different percentages of content requesters and content providers. In both scenarios, the total number of vehicles during the simulation is 608 and each vehicle, requesting a content object, issues 150 Interests during the 300 seconds simulation time.

1. In the first scenario, 25% of all vehicles request a content object (requesters) and 25% of all vehicles are content providers (producers). All other vehicles act as forwarders, i.e. they can forward Interest and Data messages according to their FIB and connect to the RSU (as described in Sections 6.2.2 and 6.3). We notice that as vehicles move in space and time, they can leave our selected area of 2km x 2km. All vehicles with content objects may leave this area during the simulation period. Hence, to avoid for the content object to disappear if all content providers leave the area, every time a vehicle enters into the selected area, we assign a 25% probability to possess the content object or a 25% probability to request this content object. We also highlight that we store the content object neither on the RSUs nor on the SDN controller.

2. With our second scenario, we wanted to evaluate the scalability of our network, by increasing the percentages of content requesters and content providers to 40%. We also assumed that a vehicle enters the area has a 40% probability to request the content object or a 40% probability to possess the content object. Vehicles that do not request or possess the content object act as forwarders. As in the first scenario, to avoid for the content object to disappear, we assign a 40% probability for a vehicle entering the selected area to request the content object, and 40% probability to possess the content object.

Table 6.3: SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Channel Frequency | 5.890e9 Hz |
| Minimum power level | -109 dBm |
| Propagation loss model | Two Ray and SimpleObstacleShadowing |
| Bit Rate | 6Mbps |
| Phy Model | IEEE 802.11p |
| Number of vehicles | 608 |
| Average Vehicle Speed | 20-30 m/s [48] |
| Area | $2km^2$ |
| Interest interval | 1s |
| Simulation time | 300s |
| GPS accuracy | ± 7.1 m |
| RBM interval | 1s |

#### Simulation Parameters

We evaluated our protocol by using the OMNET++ network simulator with the vehicular framework VEINS to support vehicular communication using the IEEE 802.11p as the physical model and SUMO to support mobility. We use the OMNET++ network simulator because in the MIC scenario we use directional antennas. OMNET++ supports directional antennas in the IEEE 802.11p standard, whereas ns-3 does not (c.f. Section 4.3.1). In OMNET++ we also customize and use a modified version of NDNOMNeT [12] with a modified version of the OpenFlow protocol [13]. Details about the evaluation parameters can be found in Table 6.3.

#### Simulation Metrics

For evaluating the proposed scheme we used the following metrics:

- *Interest Satisfaction Rate (ISR)* (c.f. Section 5.3.1).
- *Interest retransmissions* (c.f. Section 4.3.1).
- *Average hop count of an Interest* denotes the average hop count that one Interest message should travel to find a content source.
- *Number of transmitted Interests* denotes how many Interests are transmitted through the FIB towards a content source on average and how many towards

RSUs, and, hence, to the SDN controller.

- *Number of RBMs* denotes on average how many RBMs one node transmits towards the RSUs during the whole simulation.
- *Tx adaptation messages* denotes how many messages the controller sends to the RSUs for the latter to change its transmission power.
- *FIBPop messages* denotes the number of messages that the SDN controller sends to the RSUs for the latter to transmit them to vehicles to populate their FIBs.

The above metrics describe the main characteristics that we consider important in the SDN-NDN VANET for an infotainment application. Together with ISR, we measure how many times an Interest is retransmitted from requester nodes, to see how much traffic the requester nodes produce. We highlight, that the traffic produced from the requester nodes is very important, since it drives more than half of the messages existing in the network (Interest, Data, Interest transmission from the RSU to the SDN controller, SDN controllers' response with FIBPop messages to the RSU and transmission of FIBPop messages from the RSU to vehicles). Moreover, we count the average hop count of an Interest to test whether our different node configurations increase the transmission range of a vehicle and whether the SDN controller calculates a short path. Then, we analyse where the Interests are sent to test whether the SDN controller impacts the routing of messages. Finally, we measure the traffic the SDN controller produces (Tx adaptation and FIBPop) to measure its impact in the network.

### 6.4.2 Simulation Results

The simulation results are shown in Figs. 6.7–6.13. Figs. 6.7–6.13 are presented as a function of the number of installed antennas, which are the same for the vehicles and the RSUs.

Fig. 6.7a shows the average ISR of all requester nodes when 1 RSU is placed in the selected area. For the *MIC* configuration, for both percentages of requesters and providers, the ISR remains the same at around 0.95 with small fluctuations at around 0.01. We see that although the percentage of producers is higher in the second scenario, the ISR for the *MIC* configuration remains almost the same independent of the number of antennas that we install on RSU and vehicles. For the *SU-MIMO* configuration, we observe for both scenarios with 25% and 40% requesters and producers the ISR

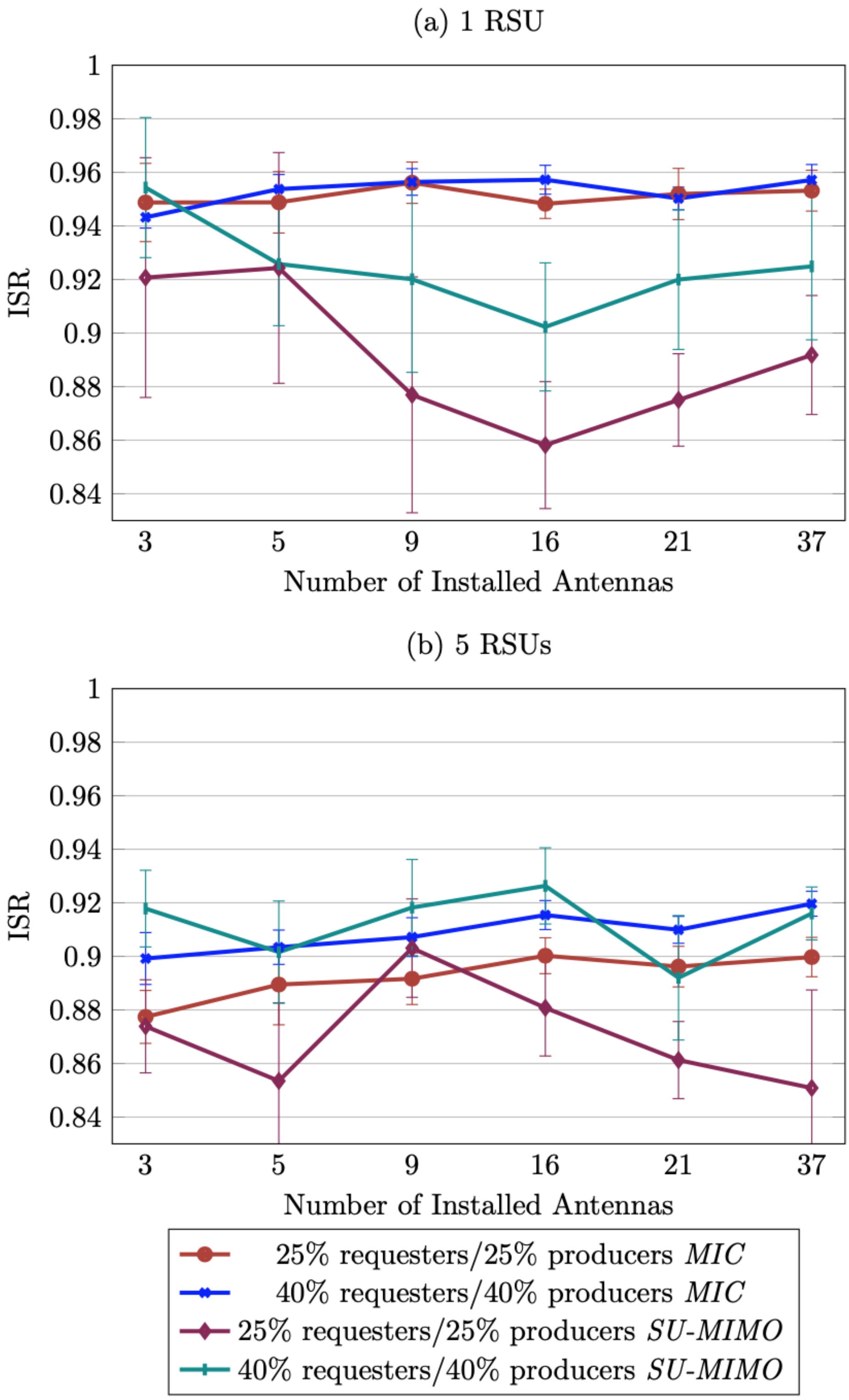



Figure 6.7: ISR in relation to number of installed antennas

drops significantly compared to all other scenarios. This is because the selected configuration is a Single-User MIMO (SU-MIMO) system. This means that a node and/or an RSU can only exchange Interests and/or content objects with one vehicle in a particular time interval. Hence, if a content provider receives many Interests from different requesters at the same time, it can only satisfy one request, leading to lower ISR. In addition, because we use a SU-MIMO system when a node steers its antennas towards a particular direction, its power towards other directions becomes smaller, leading to a smaller range and coverage area. The latter leads to high fluctuations of the ISR that is varying from 85% to 90% and is being caused because of the antenna array that we used. In our case, we placed the antennas circularly on top of a vehicle and on the RSU.

Fig. 6.7b shows the average ISR of all requester nodes when 5 RSUs are placed in the selected area. We observe that the ISR for all node configurations drops and fluctuates from 0.86 to 0.93. When more RSUs are placed, more messages exist in the network. RSUs send beacons and nodes respond with RBMs. More beacons result in higher interference, especially since we do not fine tune the transmission of the beacon messages, only the transmissions of Interests. Therefore, nodes that are connected to more than one RSUs could receive at the same time beacon messages resulting in packet loss due to bit errors.

On the other hand, Fig. 6.8a shows the Interest retransmissions when 1 RSU is placed in the 2km x 2km area. We observe that the Interest retransmissions are not the same for our scenarios. For the *MIC* configuration, we see that as the percentage of requesters increases the number of message transmissions is higher. More nodes try to retrieve content objects leading to more collisions in the same channel, leading to lost Interests. Thus, nodes need to transmit the same Interest more than once to retrieve a content object successfully. For the *SU-MIMO* configuration, the number of retransmitted Interests is high, when the ISR is low (Fig. 6.7). When a content requester does not receive the requested content object, it produces low ISR, and it retransmits the Interest. When 16 antennas are installed we observe a peak in Interest retransmissions, because with this number of antennas the ISR is very low. Low ISR means that a node sends many times the same Interest to retrieve the content object, but the Interest remains unsatisfied. On the other hand, the number of retransmissions is low when there are 5 installed antennas since the ISR is high. Moreover, Fig. 6.8b shows the Interest retransmissions when 5 RSU are placed in the area. We observe that when the ISR is low, the number of Interest retransmissions is high. A node retransmits

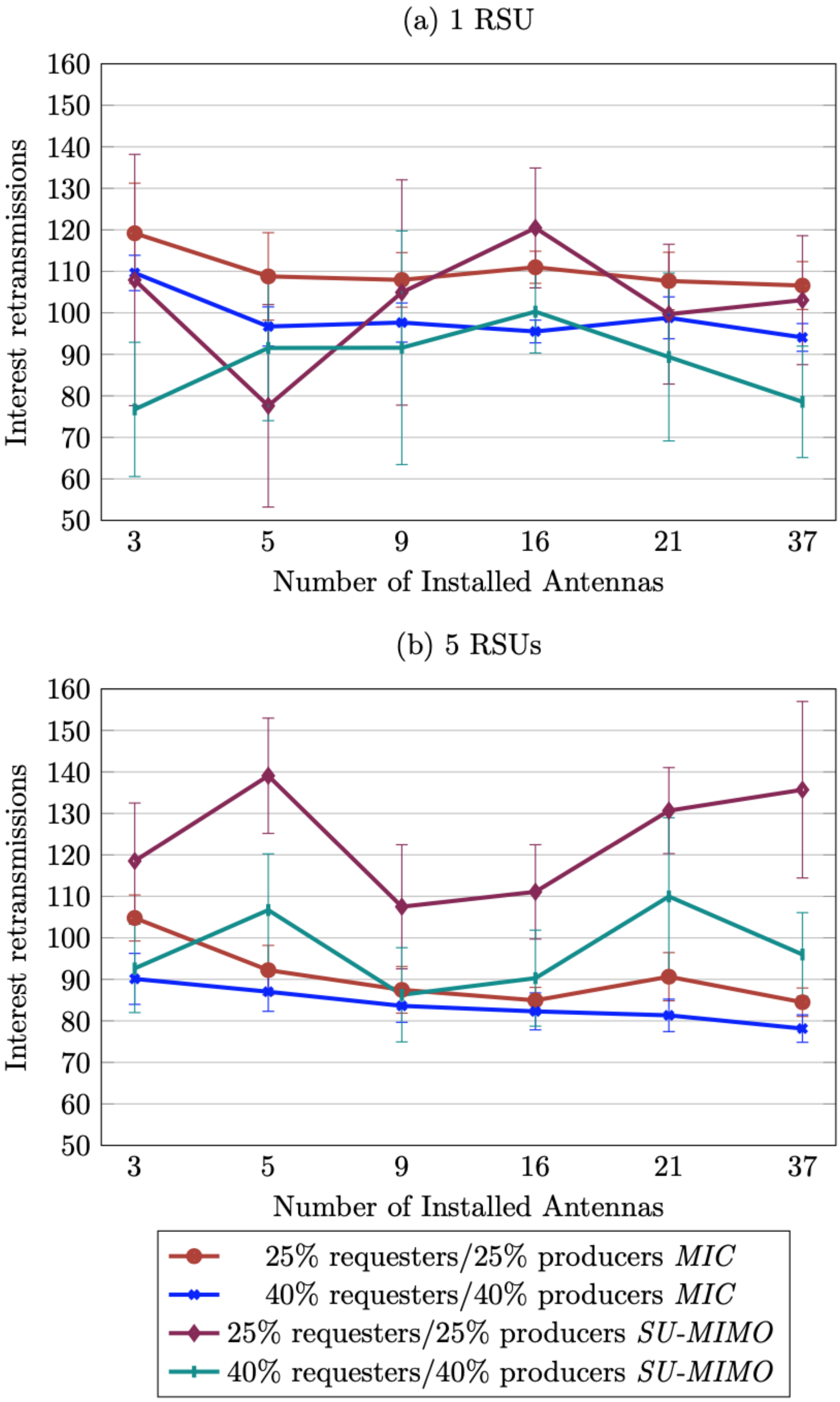



Figure 6.8: Interest Retransmissions in relation to number of installed antennas

its Interest when it is not received, leading to low ISR. Also, we show that for the *SU-MIMO* the lowest ISR occurs, when the Interest retransmissions are high compared to other scenarios and configurations.

Fig. 6.9a shows the average hop count of one Interest message when 1 RSU is placed in the selected area. The SDN controller calculates the shortest path for an Interest to travel according to the lowest number of hops. Hence, the average hop count is less than 2, meaning that on average content requesters have direct communication (1-hop communication) to the content sources. When there is no direct communication, Interests need to pass some intermediate nodes to reach a content source. Moreover, we see an increase in this hop count when the percentage of content requesters and content producers decreases. With fewer producers in the network, fewer paths exist for Interest routing and, thus, the number of hops that an Interest passes is higher. For the *MIC* configuration, we also observe a peak when 21 antennas are installed on all nodes. This directly correlates with Fig. 6.13 and will be discussed in the next paragraphs of this Chapter. For the *SU-MIMO* configuration, we observe a lower hop count compared to the *MIC* configuration. This happens because MIMO systems can increase the transmission range of a node [65]. When the transmission range is increased, nodes have higher probabilities of connecting directly to a content source, hence, the number of hops that the Interest should pass is decreased.

When installing 5 RSUs in the selected area, we show in Fig. 6.9b that the hop count increases for both configurations and scenarios and is around 2.5. This is because when a node does not have a FIB entry to the content source directly, it sends the Interest towards an RSU. If a node has a connection with many RSUs, the RSU is selected based on the highest SINR. Therefore, a node can send an Interest message to an RSU that is far away but has a better connection (higher SINR) compared to another RSU that is closer. Therefore, if the selected RSU is further away, it is more likely that a message should pass by more cars to reach a destination. We highlight that this does not affect the ISR significantly. But we can assume that the more hops a message passes (the path of the message), the more likely it is for this path to break, hence the ISR drops slightly.

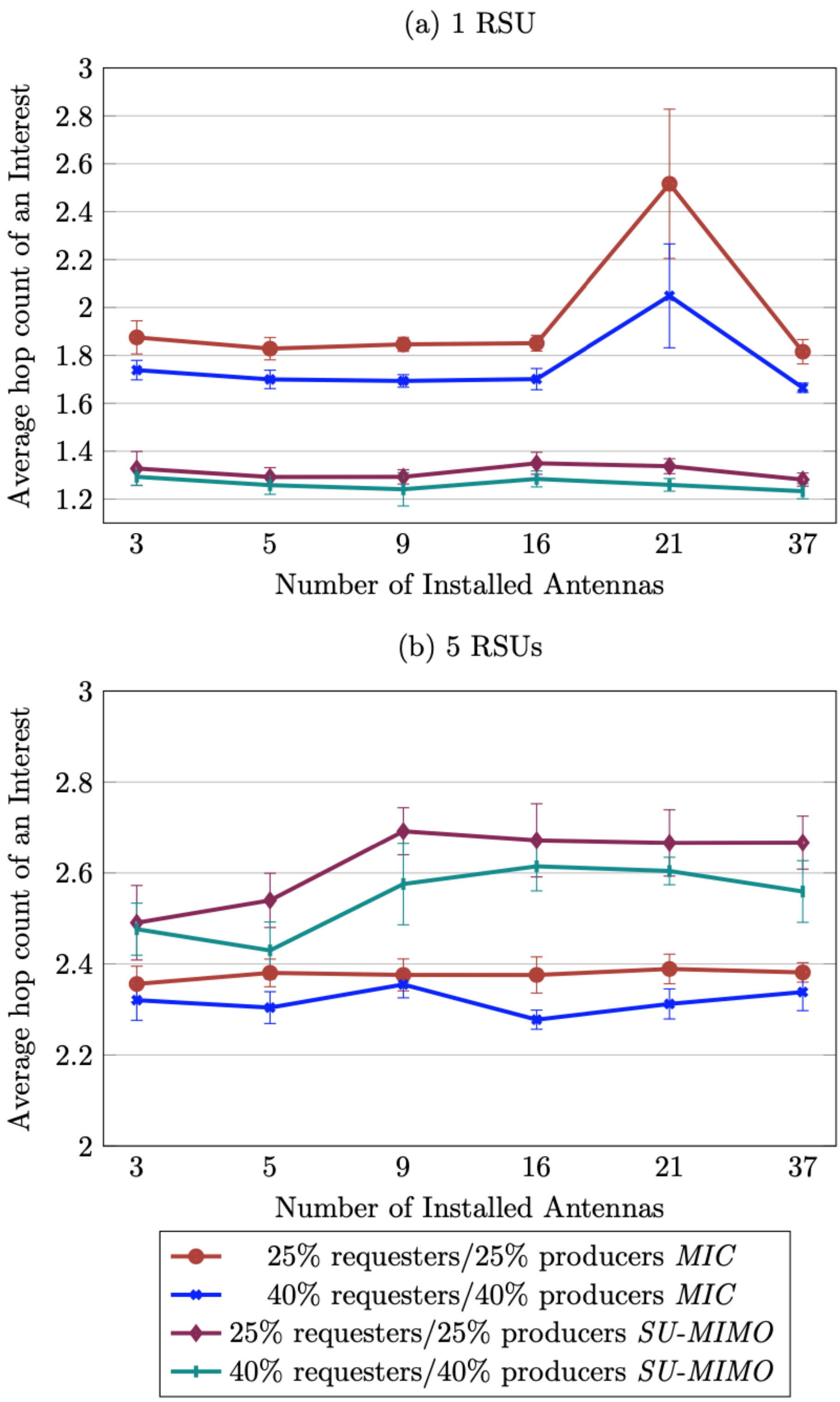



Figure 6.9: Average hop count of an Interest in relation to number of installed antennas

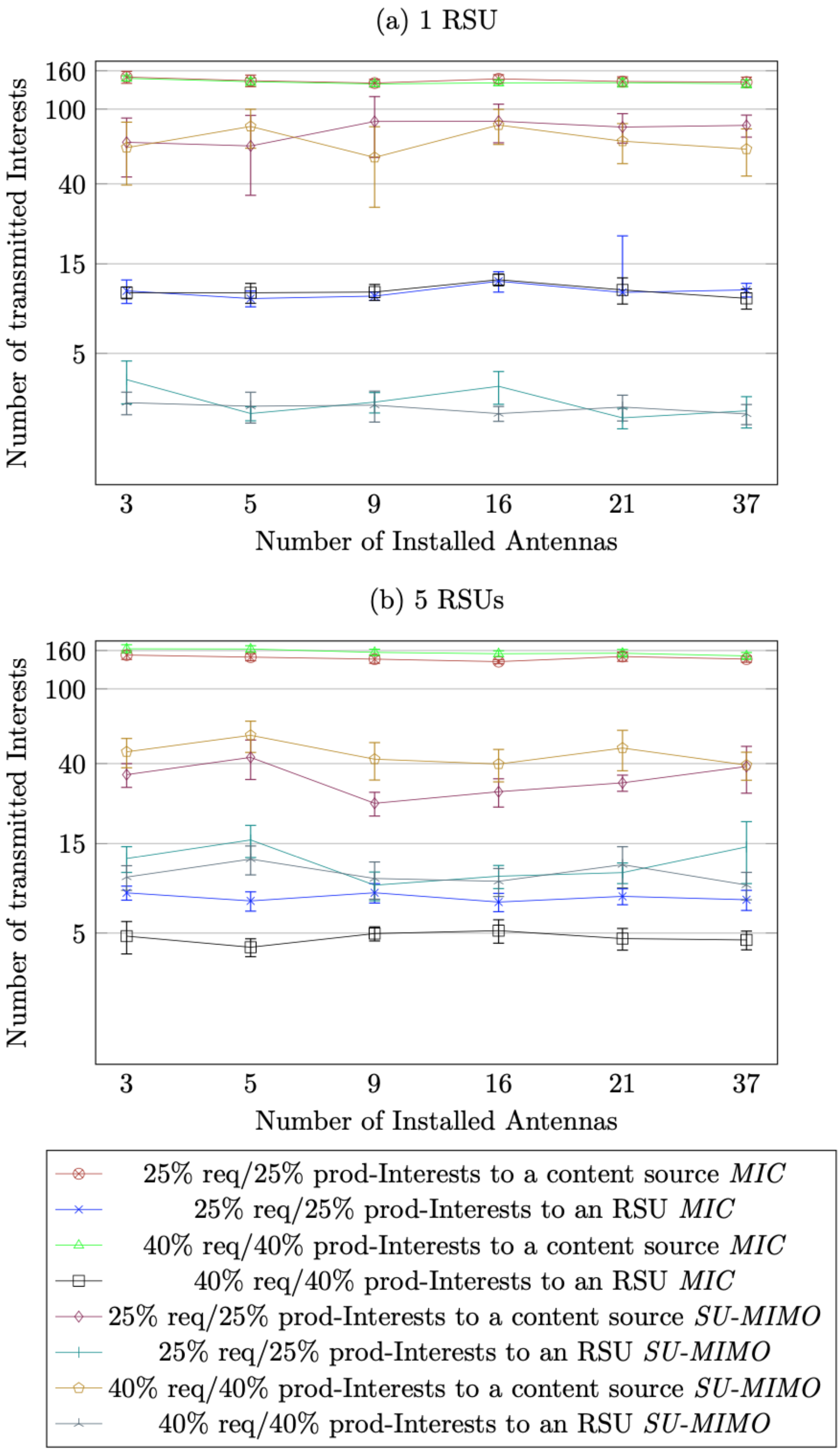



Figure 6.10: Transmitted Interests through the FIB towards a content source and an RSU in relation to number of installed antennas

Fig. 6.10a shows the average number of Interests a node sends to the RSU and to a content source when 1 RSU is placed in the selected area. A node sends an Interest to the content source when there is a FIB entry with the requested name. If there is no FIB entry towards a content source, the node sends the Interest to the RSU for the RSU to send it to the SDN controller for path calculation. For the *MIC* configuration, in average a node sends a little less than 10% of its Interests to the RSU. This shows that the calculation of the paths by the SDN controller has an impact in the network since all content retrieval related transmissions are unicast. In case of absence of the proposed infrastructure (SDN and RSUs) this 10% traffic could be directed either towards cellular interfaces or towards other nodes via broadcast transmissions. For the *SU-MIMO* configuration, we observe that the number of Interests sent towards an RSU is small, i.e. around 5%, because nodes have increased transmission range, leading to more connections to content sources. Hence, content requesters can directly transmit Interests towards a content source. This is depicted also in Fig. 6.10a, where the number of Interests towards a content source is much higher than towards an RSU. Again though, the 5% make an impact on the network, because in case of absence of the RSU and the SDN controller this traffic would be redirected towards another network, e.g. cellular. In both configurations (*MIC* and *SU-MIMO*) this traffic would burden the network by creating either huge bandwidth utilization by downloading the content object via another network (e.g. cellular) or by excessive channel utilization from the wireless broadcasts.

When 5 RSUs are placed in the selected area as shown in Fig. 6.10b, the destination of the Interests are almost the same as in Fig. 6.10a with one exception. In the *SU-MIMO* configuration, we observe that the Interests redirected towards a content source are lower when 5 RSUs are placed than when 1 RSU exists and also the Interest towards the RSUs are higher when 5 RSUs are placed compared to when 1 RSU is placed. In the *SU-MIMO* configuration, a node changes the configuration of its antennas when the node sends an Interest message. Therefore, during the content retrieval process, a requester will have steered its antennas towards a particular direction, where the content source is. If the connection to the content source breaks, the node will redirect its Interests to an RSU. When more RSUs exist, a node is more likely to have a FIB entry pointing to an RSU, therefore, it can redirect more Interests towards an RSU. Moreover, if a node does not have a connection to a content source, then it is more likely that the node will have a connection to an RSU when many RSUs are installed in its travelling area.

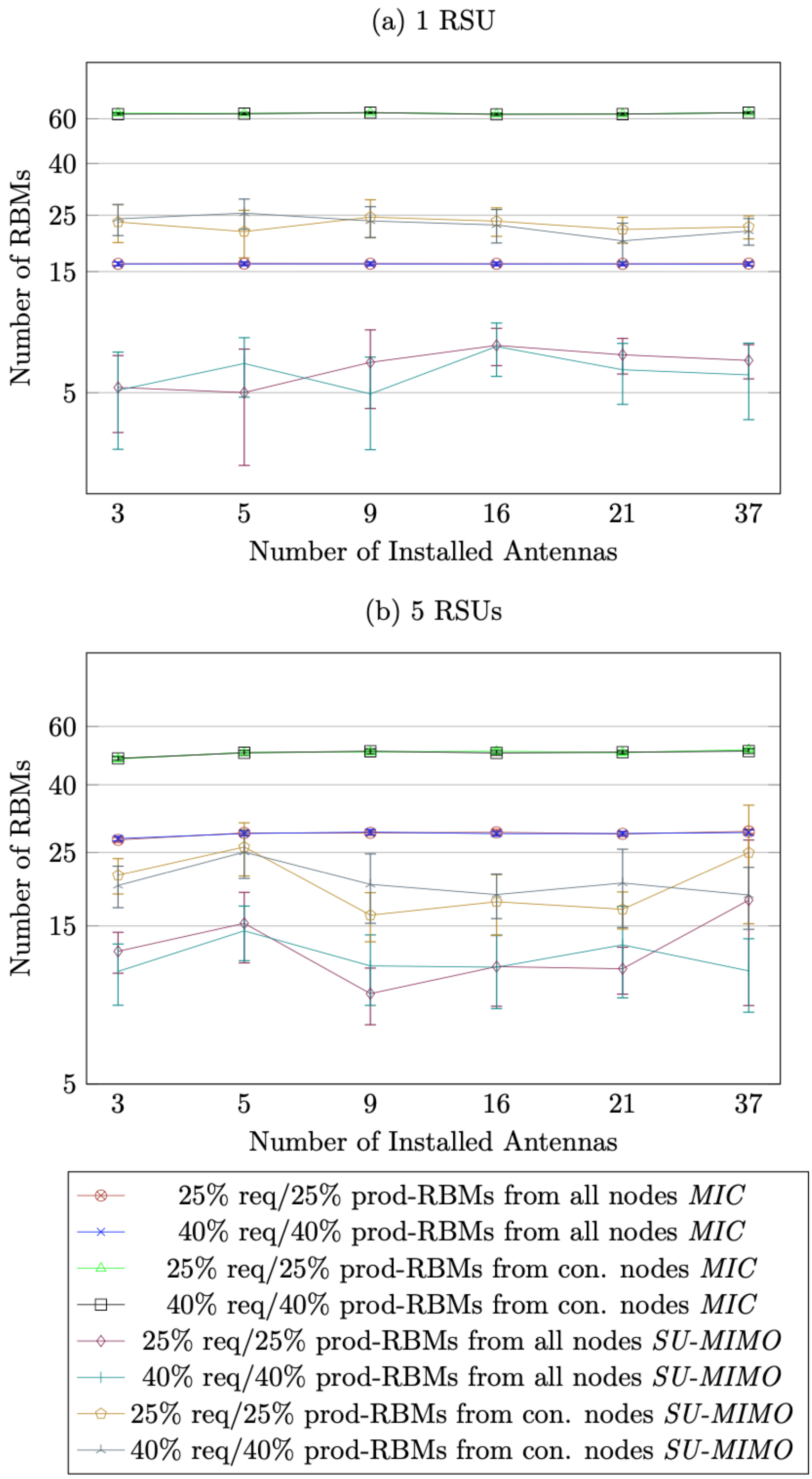



Figure 6.11: Average number of RBMs an SDN controller receives from all nodes and from only connected to all RSUs nodes in relation to number of installed antennas

Fig. 6.11 shows the number of RBMs a node sends on average. We distinguish two cases. The first case is for all network nodes. As mentioned in Section 6.2.3 the number of nodes connected to an RSU (and hence, to the SDN controller) is limited. When calculating the average number of RBMs per node, we take into account all network nodes, meaning also the ones that are not connected to the RSU. Fig. 6.11a and Fig. 6.11b show the results of our algorithm when 1 and when 5 RSUs are placed in the selected area, respectively. Fig. 6.11a shows that for the *MIC* configuration a node sends around 16 RBMs for the whole 300 seconds of communication, whereas as shown in Fig. 6.11b, for 5 RSUs a node sends around 25 RBMs. This is expected since more RSUs send more beacon messages, therefore, more nodes are connected with them. For the *SU-MIMO* configuration, a node sends around 5 RBMs when 1 RSU is installed and 15 when 5 RSUs exist, during the total simulation time. Practically, this means that for the *MIC* configuration all nodes have an active connection with the RSU for 16 seconds when one RSU is installed and 25 seconds when 5 RSUs are installed. For the *SU-MIMO* configuration, all nodes during the 300 seconds simulation time have a connection with the RSU for 5 seconds and 15 seconds when 1 and 5 RSUs are placed, respectively. The second case is when we calculate the average number of RBMs that a node directly connected to the RSU sends. When a node is directly connected to an RSU, it does not mean that there is connectivity for the whole 300 seconds of the simulation time. This means that we take into account nodes that have a stable connection with an RSU, as well as nodes that have an interrupted or short connection with an RSU. We observe that for the second case, for the *MIC* configuration, the number of RBMs per node is more than 4 times higher, around 63 messages per node when 1 RSU is placed and around 50 when 5 RSUs are placed. The connection, therefore, of every connected node is shorter with an RSU, but more nodes are connected with an RSU. For the *SU-MIMO* configuration, the number of RBMs per node is 5 times higher, at around 25 RBMs per node when 1 RSU is placed and around 20 when 5 RSUs are placed in the selected area. Even with this limited connectivity in both cases, the RSU offloads around 10% of traffic for the *MIC* configuration and 5% for the *SU-MIMO* configuration allowing the network to be more flexible in frequent mobility changes. In addition, we see that the network is scalable, meaning that the number of RBMs is almost the same for both of our scenarios.

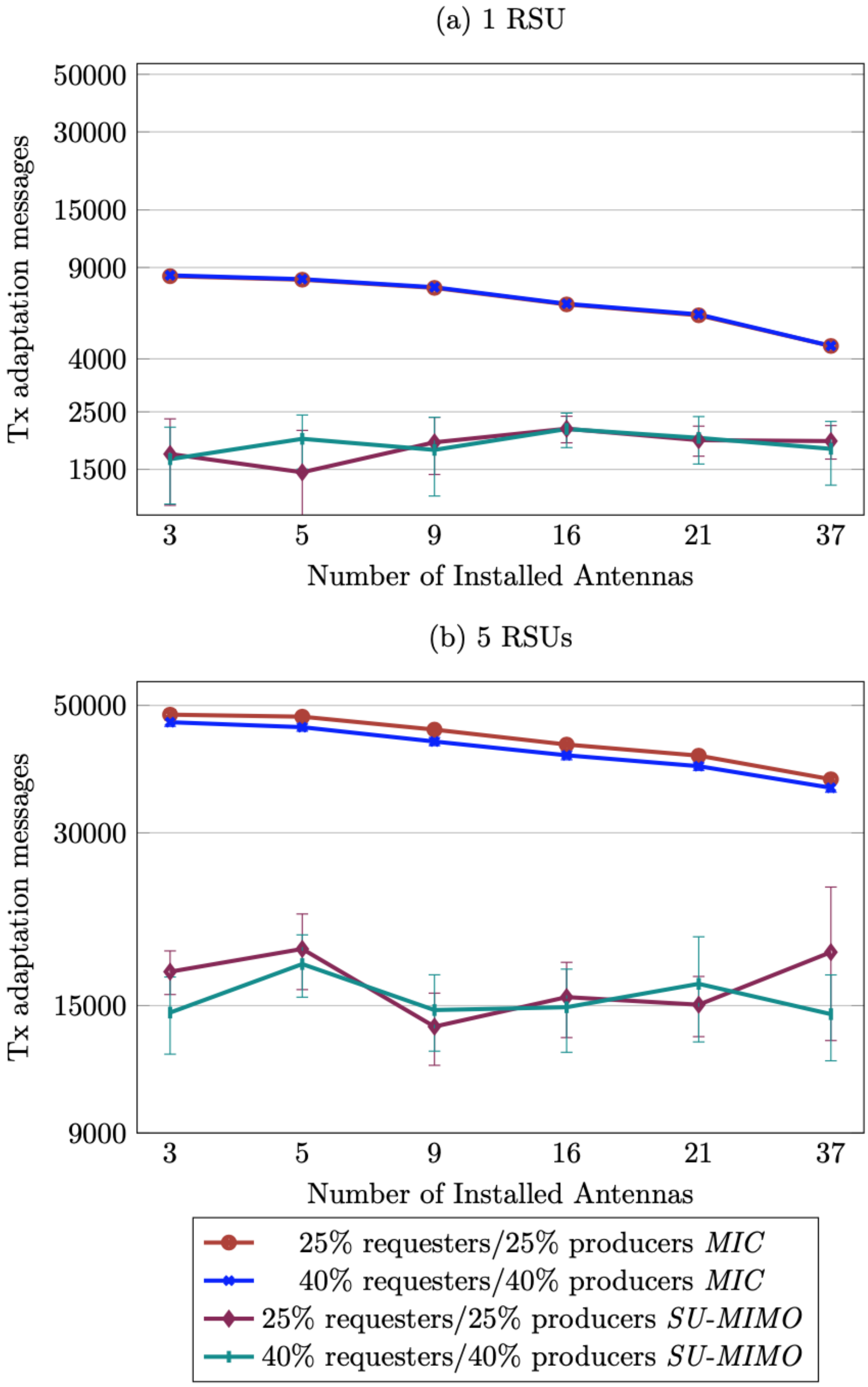



Figure 6.12: Transmission adaption messages from an SDN controller to the RSU(s) in relation to number of installed antennas

Figs. 6.12–6.13 show the traffic that the SDN controller produces and sends towards the RSUs. Fig. 6.12 shows the number of transmission power adaptation messages the SDN controller sends to an RSU (Fig. 6.12a) or to 5 RSUs (Fig. 6.12b). Using Algorithm 4 the SDN controller always tries to find a transmission power value, where the number of connected vehicles is higher than at the current time. The SDN controller sends adaptation power messages even when it knows it reached the maximum number of connected cars, to try and adapt the transmission power value, i.e. the transmission range of an RSU, to the mobile environment of our scenarios. This is because the SDN controller always tries to increase the number of connected cars to an RSU. We notice that for both when 1 RSU and 5 RSUs are placed the *MIC* configuration as the number of antennas installed in network components increases, the transmission power adaptation messages decrease. More antennas create on the one hand more interference, but on the other hand, the SDN controller learns that even if it changes the transmission power value of the RSU the number of connected cars to the RSU will not be increased. Hence, it saves resources and does not send as many transmission power adaptation messages, compared to fewer antennas installed. For the *SU-MIMO* configuration, we observe that the number of transmission adaptation messages is much lower than for the *MIC* configuration. This is because using *SU-MIMO* configuration, an RSU will steer its antennas towards different areas, making the number of cars that it communicates smaller. But, the controller is not aware of the total number of cars that can potentially connect to an RSU. Hence, it saves resources and does not instruct an RSU to change its transmission power. As expected, the number of transmission adaptation messages that an SDN controller sends is much higher, when 5 RSUs are installed compared to when 1 RSU is installed. This is because when 5 RSUs are installed the controller tries to maximize the maximum number of connected cars to all RSUs and send messages to either one when it calculates that the overall number of connected cars can increase.

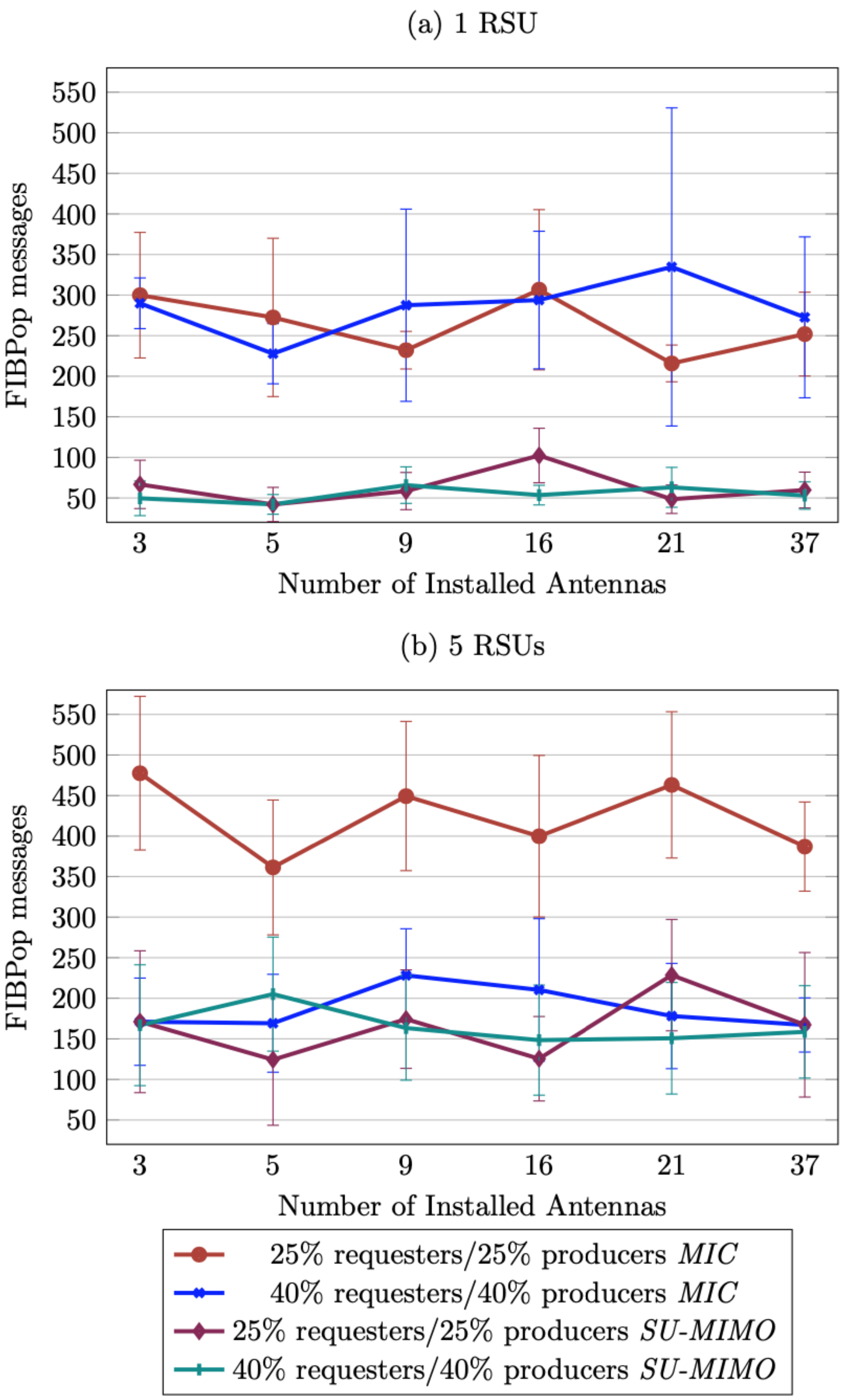



Figure 6.13: FIB population messages from the SDN controller to the RSU(s) in relation to number of installed antennas

Fig. 6.13 shows the total number of FIB population messages that the SDN controller sends to the RSUs. Each of these messages represents a FIBPop message as described in Section 6.3. For the *MIC* configuration in Fig. 6.13a, we notice a peak with high fluctuations, if 21 antennas are installed. This is because many nodes around the RSU send messages to the RSU requesting paths, and the controller responds with these paths. A node sends an Interest message to the RSU (and through the RSU to the SDN) when the node does not have a FIB entry that points to the content source. Hence, the Interest should pass through multiple hops to reach the content source. Therefore, this peak in Fig. 6.13 is correlated with Fig. 6.9, where we observe that the hop count of an Interest message also peaks when 21 antennas are installed. Since the SDN controller sends more FIB adaptation messages, the requesters do not have a direct connection to a content source (1-hop communication). Hence, the Interest should pass through multiple nodes to be satisfied. For the *SU-MIMO* configuration in Fig. 6.13a we observe that the number of FIBPop messages that the controller sends is much smaller than for the *MIC* configuration. This is because using MIMO nodes can increase their transmission ranges, reaching content sources directly. This is highly correlated with the results depicted in Fig. 6.9 and Fig. 6.10, where the hop count of an Interest message is small leading to the number of transmitted Interests towards the RSU to be small. Hence, when the SDN controller receives fewer Interests from the RSU, it will create less FIBPop messages. In Fig. 6.13b, we present the number of FIBPop messages from the controller to all RSUs. First, we show that for the *MIC* configuration the number of FIBPop messages is higher than in Fig. 6.13a, when 25% of nodes request the content object. This happens also for the *SU-MIMO* configuration. More RSUs result in more interference in the network, meaning that more messages are lost, leading to fewer nodes to have a direct connection to a content source. Hence, more nodes that are now connected to an RSU (Fig. 6.11b) redirect their messages towards an RSU. Moreover, higher hop count means that the path that the SDN controller calculates, requires more FIBPop adaptation messages. Moreover, for the *MIC* configuration, when 40% of nodes request the content object, the number of FIBPop messages drops compared to Fig. 6.13a. This is because more RSUs will connect more nodes to the SDN controller. At the same time, more RSUs transmit more beacon messages. When content sources respond with RBMs to the RSUs, as described in Section 6.2.2, these RBMs contain the content names of the content object the content sources have. Nodes overhear this transmission and enter an entry into their FIB indicating that they have a connection to a content source. Therefore, more RSUs send more beacons, reaching more content sources, triggering more RBMs, resulting in nodes to enter more FIB entries into their FIBs (that point towards a content source).

## 6.5 Conclusions

Software Defined Networking (SDN) has made a huge impact on modern network architectures by making networks components programmable. We use SDN in an NDN-VANET, where a content object is retrieved based on its name for decoupling host and content location. We study the impact of SDN when network nodes request content objects, without broadcasting neither requests nor content objects. Hence, we avoid broadcast transmissions and use them only for broadcasting 1 hop beacon messages as defined in the IEEE 802.11p standard. We show that SDN assists in the scalability of the vehicular network, as more requests are issued into the network, its performance is not affected. We also support path breaks, by making the SDN controller responsible for finding routing paths and populate the routing tables of vehicles.

We also study two different node configurations for reducing the spreading area of a message. In particular, for the first configuration, we use both directional and omnidirectional antennas installed in vehicles and RSUs. The communication between the RSUs and the vehicles is performed via the omnidirectional antenna and the V2V communication for content retrieval is performed via the directional antennas. For the second node configuration, we use only omnidirectional antennas in vehicles and RSUs. Then, we use these antennas as a MIMO system to target their energy in a particular direction.

The presented results show that content requesters on average have direct communication with content sources. But when this is not the case, 5% to 10% of traffic, when using different node configurations, is sent to an SDN controller instead of broadcasting these messages to the network. In addition, nodes retrieve from 85% to 95% on average of their requested content object, depending on the different node configurations. The SDN controller adapts the transmission power of an RSU successfully, allowing it to connect with as many nodes as possible.

# 7 Conclusions and Outlook

## 7.1 Main Contributions

The main goal of this thesis is to answer the research question that were posed in Chapter 1.2. Table 7.1 presents the research question and our proposed solutions to solve them. In this Section, we summarize the contributions of this thesis.

The first research question (**RQ1**) addressed in the thesis is how to reduce the number of broadcast transmissions of messages in VANETs when the number of interconnected cars is high (Section 1.2.1). Broadcast transmissions lead to waste of resources since unnecessary transmissions occupy the channel and reduce the bandwidth of the network. Moreover, broadcasting on the same channel leads to message collisions. To answer **RQ1**, in Chapter 3 we presented two routing protocols for NDN-VANETs, named Multihop, Multipath and Multichannel routing protocol for NDN-VANETs (*MMM-VNDN*) and improved MMM-VNDN, *iMMM-VNDN*. Both of these protocols introduce multihop communication between a requester vehicle and a content source. In MMM-VNDN we broadcast every message. We include two new fields into the NDN messages. These fields contain MAC addresses that

Table 7.1: Research Questions and Answers

| RESEARCH QUESTIONS | ANSWERS |
|---|---|
| How to reduce the number of broadcast transmissions of messages in VANETs when the number of interconnected cars is high? (Section 1.2.1) | We introduce new routing protocols for NDN-VANETs that unicast messages. The unicast transmissions of messages are based on different next hop selection techniques. (Chapter 3) |
| How to limit the dissemination area of transmitted messages in VANETs when the number of interconnected cars is high? (Section 1.2.2) | We propose using multiple directional antennas to support simultaneous message transmissions towards particular directions. We develop a routing protocol to choose the appropriate antenna based on the position of nodes. (Chapter 4) |
| How deployed infrastructure combined with the integration of ICN and appropriate routing protocols assist content retrieval in VANETs? (Section 1.2.3) | We develop routing protocols using Road Side Units (RSUs) as deployed infrastructure. RSUs participate in the content exchange process, acting as a gateway, when a requester node cannot retrieve the requested content object through V2V communication. (Chapter 5) |
| Does one centralized architecture combined with the integration of ICN, improves network performance, in terms of vehicular connectivity and content retrieval, in high density VANETs? (Section 1.2.4) | We propose using Software Defined Networking (SDN) as a centralized solution for VANETs. SDN enables performing all routing decisions in the SDN controller. The SDN controller calculates paths, to avoid the unnecessary occupation of the channel as well as to save network resources. (Chapter 6) |

point to the previous node and the next node of the message. Based on these fields nodes accept or reject incoming messages. When nodes reject incoming messages, the number of broadcast transmissions of messages is reduced. In iMMM-VNDN, transmissions can be either broadcast or unicast. In both of these protocols, we introduce two new fields in the node Data structures, i.e. FIB, PIT and CS. These fields assist the routing of Interest and Data messages towards a particular node. Then, we introduce three strategies to choose the best path from the requester to the content source. This selection is performed in every node that receives a message, i.e. every node decides according to its routing table (FIB) which is the next node that should receive the message. The first strategy instructs every node to choose the next hop in a round robin manner. Hence, we distribute the traffic uniformly to all nodes in the network. The second strategy is to choose the next node based on the communication latency. The latency is defined as the time passed from the time a node sent an Interest message to the time that the node received the corresponding Data message. This strategy allows reducing the latency of the requester during the content retrieval process. The third next hop selection strategy is the combination of the above. A node chooses the least used next hop. If next hops in the routing table have been used the same number of times, the node selects the one with the lowest latency. Hence, we distribute the traffic uniformly and at the same time we reduce the latency. Our results show that MMM-VNDN outperforms the flooding strategy by rejecting incoming messages and constructing paths based on the MAC addresses of nodes. In addition, iMMM-VNDN outperforms MMM-VNDN, the flooding strategy and other state of the art routing protocols in terms of Interest Satisfaction Rate (ISR). iMMM-VNDN also reduces the average latency of up to 12 seconds in the requester node while keeping average jitter less than 1 ms. Therefore, iMMM-VNDN is an effective solution for applications requiring high ISR, low latency and jitter.

The second research question (**RQ2**) addressed in the thesis is how to limit the dissemination area of transmitted messages in VANETs when the number of interconnected cars is high (Section 1.2.2). When a message is transmitted through a wireless device, the direction of the electromagnetic waves is determined by the type of antennas installed in vehicles or by techniques to steer these antennas towards the desired location. Usually, vehicles and Wi-Fi routers have omnidirectional antennas installed, meaning that the signal is transmitted towards all directions with the same strength. This omnidirectional transmission can lead to the redundant occupation of the channel. When the channel is busy, nodes can experience large delays competing for channel occupation. Therefore, in Chapter 4 we limit the dissemination area of messages by installing directional antennas in vehicles and using V2V communication. We presented a routing protocol called enhanced Geographical Aware Routing Protocol (*eGaRP*) for NDN-VANETs. We designed eGaRP for small local areas inside cities, where communication between vehicles can be disruptive. eGaRP introduces V2V directional communication between vehicles, by installing directional antennas in each vehicle. Consequently, nodes can send messages towards particular directions reducing the channel occupation of vehicles outside of the spreading area of messages. In addition, to establish a better connection between vehicles, we mechanically rotate the selected antenna of communication by a particular angle to point to the location of the vehicle that we want to send a message. By this rotation, we allow stronger signal and better connection between the two nodes. Our results indicate that eGaRP outperforms the iMMM-VNDN routing protocol when applied in city centres during rush hours. Specifically, the number of delivered Data messages in the requester node is increased up to 40%, the average latency is reduced up to 100 ms, and the number of request retransmissions are decreased by 9 times in the requester node. Therefore, eGaRP is an advanced solution that further improves iMMM-VNDN by providing better results in the overall content retrieval process using only V2V communication.

The third research question (**RQ3**) addressed in the thesis is how deployed infrastructure with ICN and appropriate routing protocols assist content retrieval in VANETs (Section 1.2.3). To answer RQ3, in Chapter 5 we present two routing strategies that use infrastructure, in particular Road Side Units (RSUs), as a network component in the NDN routing process. The proposed routing strategies consist on two phases. The first phase is called learning phase and it is based on beacon transmission between vehicles and the message transmissions between vehicles and infrastructure. In the learning phase an RSU periodically broadcasts beacon messages. Nodes receiving these beacon messages respond with a Respond to Beacon Message (RBM) message. RBMs assist the RSU to identify the content sources that exist around it. During the learning phase, when the content sources broadcast RBMs, they include the name of the content object they produce or hold in their CSs. Therefore, when nodes and the RSU in the vicinity of content sources receive RBMs, they enter into their FIB the content object name and the MAC address of the content source. In addition, all vehicles during the learning phase identify their neighbours, i.e. other vehicles that are connected with them. In the second phase, called forwarding phase, a node requests a content object. For the content retrieval process, we developed two approaches. In the first, named *linked approach*, the requester always sends its request to the RSU. The RSU already knows a content source (from the learning phase) and, hence, it can transmit directly this request to the content source. The content source responds with the content object to the RSU and the latter sends it to the requester node. In the second approach, called *hybrid approach*, the requester checks its FIB to identify if there is an entry with the requested content name. If such an entry exists, the requester unicasts its request to the node specified in the FIB entry. If there is not such a FIB entry, the requester sends its request to the RSU. Hence, we use the RSU as a backup mechanism when the requester cannot retrieve the requested content object through V2V communication. Our results indicate that in the linked approach many requests remain unsatisfied. In particular, when the number of vehicles is high, the linked approach does not deliver any messages to the network, because we create congestion around the RSU when too many messages compete for the same resources. With the hybrid approach, we outperform our previous routing protocol iMMM-VNDN, the flooding strategy and the AODV routing protocol [117]. In particular, the number of Delivered Data messages is around 5 times higher than with the other protocols, the ISR increases up to 3 times more and the latency is kept lower than the other approaches around 25 ms. Therefore, we highlight that using infrastructure, and in particular RSUs as a main network component, fails because congestion is created, especially with a large number of cars. On the other hand, we highlight that using RSUs as a back-up mechanism in VANETs to assist the content retrieval process improves network performance compared with other V2V routing algorithms.

Finally, the fourth research question (**RQ4**) addressed in the thesis is whether a centralized architecture combined with the integration of ICN improves network performance in high density VANETs (Section 1.2.4). To answer RQ4, in Chapter 6 we use Software Defined Networking (SDN) architecture to centralize the network and we investigate the impact of SDN in an NDN-VANET. We deploy an SDN controller outside of city centres that is connected with RSUs, which act as switches. We assume RSUs are already deployed in city roads. Then, first, we deal with the heavy traffic that is created around an RSU, when many vehicles try to connect to it, by adapting its transmission power. This adaptation allows for an RSU to change its range, when necessary. The SDN controller gathers network information by cars connected an RSU and is responsible for instructing the RSU to change its range according to the connected vehicles. This prevents collisions and broadcast storms around the RSU. Moreover, we employ the native SDN functionality, which is to perform routing decisions. Vehicles requesting content objects send their request to the RSU, and the latter sends it to the SDN controller. The SDN controller, then, is responsible for identifying paths between nodes and to send messages to the vehicles participating in the content exchange to populate their routing tables. Thus, vehicles that participate in the content retrieval process, have entries in their FIBs and unicast messages. We experiment with two different node configurations, both for vehicles and RSUs. In the first, we install on them an omnidirectional and many directional antennas. Communication between vehicles and RSU is performed using the installed omnidirectional antenna. Content retrieval is performed by using one directional antenna that points to the desired location. In the second configuration, we install multiple omnidirectional antennas in vehicles and RSUs. The communication is performed via these antennas, that we use as a MIMO system. Our results show that by using SDN to deal with connectivity problems in the VANET, we keep the ISR around 0.9, independent of the networking and vehicular traffic. We also show that 5%-10% of network traffic is directed towards the SDN controller that makes the routing decisions. Thus, the usage of SDN highlights the efficiency of our proposed solution, since we avoid broadcasting messages and instead we save network resources by sending 5% to 10% of traffic to an SDN controller.

## 7.2 Future Work

Although VANETs have been introduced since 2001 [139], as communication technologies advance, so should the communication solutions for vehicular environments. In this thesis, we propose communication techniques and routing algorithms when using NDN for VANETs. Nevertheless, there are still enhancements that can improve their application performance or adjustments to cope with the requirements of specific applications.

A vehicle should be autonomous and rely solely on its own knowledge about its surrounding environment. Therefore, the need for efficient schemes that use only what a vehicle could know is necessary. Studies have been focusing on utilizing Multiple-User Multiple-Input-Multiple-Output (MU-MIMO) techniques in wireless communications [75, 130] to improve channel performance in both MANETs and VANETs. The addition of MU-MIMO techniques in the vehicular communication scheme enables the communication of a base station with multiple users, increases

Table 7.2: Comparison of IEEE 802.11p and IEEE 802.11bd [112]

| Feature | IEEE 802.11p | IEEE 802.11 db |
|---|---|---|
| Radio bands of operation | 5.9 GHz | 5.9 GHz & 60GHz |
| Channel coding | BCC | LDPC |
| Re-transmissions | None | Congestion avoidance |
| Countermeasures against Doppler shift | Node | Midambles |
| Sub-carrier spacing | 156.25 kHz | 312.5 kHz, 156.25 kHz, 78.125 kHz |
| Supported relative speeds | 252 kmph | 500 kmph |
| Spatial streams | One | Multiple |

the utilized bandwidth, as well as the channel capacity.

Wireless LANs, such as the most recent 802.11ac support the SU-MIMO approach [37]. However, the current IEEE 802.11p protocol that is used for vehicular communication, does not provide a specialized version for MIMO. The results of enabling SU-MIMO in VANETs have been studied in the literature over the last years and are promising [89, 111]. Nevertheless, some remarks should be made:

- The MIMO Physical layer model should be developed in detail for the current standard used in VANETs.
- Channel access control functions should be properly designed for improved system performance, especially in highly dense situations, when many nodes are competing for channel access.
- When increasing the bandwidth, larger packets can be forwarded without excessive retransmissions. The packet is divided in fewer chunks and these can be transmitted probably in a noisier channel.

Moreover, a new standard for V2X connectivity is currently being developed by the 802.11 task group, named IEEE 802.11bd. The goal of IEEE 802.11bd is to double the throughput of vehicular networks, support high speeds (up to 250 km/h), improve transmission range by having at least one module that achieves twice the communication range of the IEEE 802.11p standard, and reduce channel collisions [33]. The IEEE 802.11bd is expected to increase the data rates it can achieve by adopting some of the existing PHY technologies such as LDPC, MIMO, 256 QAM modulation, and 20 MHz bandwidth [33].

Furthermore, another interesting line of work refers to the interoperability between the two standards, i.e. the IEEE 802.11bd and the IEEE 802.11p. For instance, the IEEE 802.11p devices shall be compatible with the IEEE 802.11bd devices by decoding (at least one mode of) the transmissions from 802.11bd devices, and vice-versa. To achieve compatibility between the different standards, IEEE 802.11p and IEEE 802.11bd, the packet format in IEEE 802.11db changes by including some fields from the IEEE 802.11p packet format together with new ones [112]. To summarize, the key differences between the two standards are shown in Table 7.2. Therefore, the introduction of IEEE 802.11bd is expected to improve the reliability and stability of vehicular applications.

Finally, the integration of DTN and ICN in VANETs can increase network performance and reliability. DTN can bind different internetworks and incorporate devices and applications with limited and/or local functionality, which require a form of internetworking capability [125]. The core advantage of DTN is that it supports the store-carry and forward mechanism, meaning that nodes can store a packet in their cache, carry the packet and forward it to the destination. This can improve the connectivity in a VANET, and reduce the overall number of message transmissions since vehicles could act as agents to transfer packets from one location to another. Furthermore, DTN supports custody transfer of a message: A node receiving a message should assure that this message will reach its destination. This offers many advantages in NDN, where the requester node is the one deciding on whether to retransmit a request. Custody transfer could offer flexibility in NDN-VANETs, reduce the number of retransmitted messages from the requester node, and, therefore, reduce the overall number of messages in the network. The integration of DTN with NDN for opportunistic environments has been proposed in the UMOBILE architecture [9]. One significant challenge that needs to be addressed, though, is that forwarding is performed in the DTN layer. This will violate a main NDN design principle that Data packets always follow the PIT entries of nodes to be forwarded to the requester node.

As a next step, an architecture integrating NDN and DTN in a centralized way, i.e. using SDN, could be considered. This architecture could exploit the core advantages of both approaches and cohesively integrate them:

- SDN could perform PIT population to nodes, where a Data message should pass to reach the requester node. This would allow routing to be performed in the NDN layer, without violating NDN principles.
- SDN could collect traces from vehicles and find their mobility patterns. Then, an SDN controller could take all routing decisions (multihop path construction) based on the application requirements and the mobility of nodes.
- SDN could be used to install critical entries to vehicles when the message is urgent. For instance, if an agent carrying a Data packet with a small lifetime (e.g. video feed of a traffic camera) becomes unavailable, the SDN could be immediately notified and act according to the application requirements.
- SDN could decide on whether a message should be transmitted via the DTN layer or the NDN layer. This separation is important since applications define their delay-tolerance. For instance, an application, such as a map of a city, could be delay-tolerant and could be performed in the DTN layer. On the other hand, a critical application or an application with a small lifetime, such as finding an optimal travel route for a geographical destination, depends on the gathered information (e.g. traffic congestion, closed roads, traffic lights signalling, etc). This information is only valid for a short duration, therefore, utilizing the NDN layer could be considered a more efficient solution.

Taking into account the above, albeit the improved physical layer of the vehicular nodes, the integration of ICN with real-life applications of VANETs is still an open issue. But with improvements and combination with other technologies (such as DTN, SDN), ICN in VANETs could be an efficient, applicable solution in the near future.

# Bibliography


[1] "Gartner predicts by 2020, a quarter billion connected vehicles," http://www.gartner.com/newsroom/id/2970017, accessed February 2020.

[2] "Internet History OF 1980s," https://www.computerhistory.org/internethistory/1980s/, accessed October 2020.

[3] "Internet History OF 1980s," https://sciencenode.org/feature/How%20did%20smartphones%20evolve.php, accessed October 2020.

[4] "NCC Strategy," https://www.nsnam.org/doxygen/classns3_1_1_three_log_distance_propagation_loss_model.html, accessed September 2017.

[5] "NCC Strategy," https://www.nsnam.org/doxygen/classns3_1_1_nakagami_propagation_loss_model.html, accessed September 2017.

[6] "NCC Strategy," https://www.nsnam.org/doxygen/classns3_1_1_two_ray_ground_propagation_loss_model.html, accessed September 2017.

[7] "NCC Strategy," https://redmine.named-data.net/projects/nfd/wiki/CcndStrategy, accessed September 2017.

[8] "The ns-3 Network Simulator," https://www.nsnam.org, accessed September 2017.

[9] "UMOBILE-Universal, mobile-centric and opportunistic communications architecture," http://umobile-project.eu/overview, accessed October 2020.

[10] "Fp7-comet project," http://www.comet-project.org/, 2010 (Accessed March 2020).

[11] “IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Identifiers,” *IEEE Std. 1609.12-2019*, 2019.

[12] “An ndn framework for omnet++,” "https://github.com/amar-ox/NDNOMNeT", Retrieved 2020.

[13] “An openflow extension for the omnet++ inet framework,” "https://github.com/lsinfo3/ofomnet", Retrieved 2020.

[14] M. Aazam, K. A. Harras, and S. Zeadally, “Fog computing for 5g tactile industrial internet of things: Qoe-aware resource allocation model,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3085–3092, 2019.

[15] N. Abani, T. Braun, and M. Gerla, “Proactive caching with mobility prediction under uncertainty in information-centric networks,” in *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 88–97.

[16] W. Abdou, B. Darties, and N. Mbarek, “Priority levels based multi-hop broadcasting method for vehicular ad hoc networks,” *annals of telecommunications-annales des télécommunications*, vol. 70, no. 7-8, pp. 359–368, 2015.

[17] A. Aboud, H. Touati, and B. Hnich, “Efficient forwarding strategy in a ndn-based internet of things,” *Cluster Computing*, vol. 22, no. 3, pp. 805–818, 2019.

[18] A. Afanasyev, I. Moiseenko, and L. Zhang. (2012, October) ndnsim: Ndn simulator for ns-3. [Online]. Available: https://named-data.net/wp-content/uploads/TRndnsim.pdf

[19] A. Afanasyev, J. Shi, B. Zhang, L. Zhang *et al.* (2016, October) Nfd developer’s guide. [Online]. Available: https://named-data.net/wp-content/uploads/2016/10/ndn-0021-7-nfd-developer-guide.pdf

[20] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, D. Kim, and M. Gerla, “Conet: Controlled data packets propagation in vehicular named data networks,” in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 620–625.

[21] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, D. Kim, H. Song, and J. Lloret, “Codie: Controlled data and interest evaluation in vehicular named data networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3954–3963, 2016.

[22] A. B. Al-Khalil, A. Al-Sherbaz, and S. J. Turner, "Enhancing the physical layer in v2v communication using ofdm-mimo techniques," 2013.

[23] A. Aliyu, A. H. Abdullah, O. Kaiwartya, Y. Cao, M. J. Usman, S. Kumar, D. Lobiyal, and R. S. Raw, "Cloud computing in vanets: architecture, taxonomy, and challenges," *IETE Technical Review*, vol. 35, no. 5, pp. 523–547, 2018.

[24] M. Amadeo, C. Campolo, and A. Molinaro, "Content-centric networking: is that a solution for upcoming vehicular networks?" in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications.* ACM, 2012, pp. 99–102.

[25] ——, "Content-centric vehicular networking: An evaluation study," in *2012 Third International Conference on The Network of the Future (NOF)*, Nov 2012, pp. 1–5.

[26] ——, "Enhancing content-centric networking for vehicular environments," *Computer Networks*, vol. 57, no. 16, pp. 3222 – 3234, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613002211

[27] ——, "Crown: Content-centric networking in vehicular ad hoc networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1380–1383, 2012.

[28] ——, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *Journal of Network and Computer Applications*, vol. 50, pp. 148–158, 2015.

[29] C. Anastasiades, J. Weber, and T. Braun, "Dynamic unicast: Information-centric multi-hop routing for mobile ad-hoc networks," *Computer Networks*, vol. 107, pp. 208–219, 2016.

[30] C. Anastasiades and T. Braun, "Information-centric communication in mobile and wireless networks," Ph.D. dissertation, Universität Bern, 2016.

[31] C. Anastasiades, W. E. M. El Alami, and T. Braun, "Agent-based content retrieval for opportunistic content-centric networks," in *International Conference on Wired/Wireless Internet Communications.* Springer, 2014, pp. 175–188.

[32] C. Anastasiades, A. Uruqi, and T. Braun, "Content discovery in opportunistic content-centric networks," in *37th Annual IEEE Conference on Local Computer Networks-Workshops.* IEEE, 2012, pp. 1044–1052.

[33] W. Anwar, N. Franchi, and G. Fettweis, "Physical layer evaluation of v2x communications technologies: 5g nr-v2x, lte-v2x, ieee 802.11 bd, and ieee 802.11 p," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–7.

[34] E. A. M. Association, "Acea report vehicles in use europe 2018," https://www.acea.be/uploads/statistic_documents/ACEA_Report_Vehicles_in_use-Europe_2018.pdf, 2018 (Accessed April 2019).

[35] E. Aubry, T. Silverston, and I. Chrisment, "Srsc: Sdn-based routing scheme for ccn," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2015, pp. 1–5.

[36] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[37] O. Bejarano, E. W. Knightly, and M. Park, "Ieee 802.11 ac: from channelization to multi-user mimo," *IEEE Communications Magazine*, vol. 51, no. 10, pp. 84–90, 2013.

[38] S. K. Bhoi and P. M. Khilar, "Vehicular communication: a survey," *IET Networks*, vol. 3, no. 3, pp. 204–217, 2013.

[39] C. A. Brennand, F. D. da Cunha, G. Maia, E. Cerqueira, A. A. Loureiro, and L. A. Villas, "Fox: A traffic management system of computer-based vehicles fog," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 982–987.

[40] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papali, "Multipath congestion control in content-centric networks," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 363–368.

[41] B.-J. Chang and J.-M. Chiou, "Cloud computing-based analyses to predict vehicle driving shockwave for active safe driving in intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[42] M. Charitos and G. Kalivas, "Mimo hetnet ieee 802.11 p–lte deployment in a vehicular urban environment," *Vehicular Communications*, vol. 9, pp. 222–232, 2017.

[43] S. Charpinel, C. A. S. Santos, A. B. Vieira, R. Villaca, and M. Martinello, "Sdccn: A novel software defined content-centric networking approach," in *2016 IEEE 30Th international conference on advanced information networking and applications (AINA)*. IEEE, 2016, pp. 87–94.

[44] M. Chen, D. O. Mau, Y. Zhang, T. Taleb, and V. C. Leung, "Vendnet: Vehicular named data network," *Vehicular Communications*, vol. 1, no. 4, pp. 208–213, 2014.

[45] D. R. Cheriton and M. Gritter, "Triad: A new next-generation internet architecture," 2000.

[46] Cisco, "Antenna patterns and their meaning," https://www.industrialnetworking.com/pdf/Antenna-Patterns.pdf, 2007 (Accessed August 2020).

[47] L. Codeca, R. Frank, and T. Engel, "Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research," in *Vehicular Networking Conference (VNC), 2015 IEEE*. IEEE, 2015, pp. 1–8.

[48] L. Codeca, R. Frank, S. Faye, and T. Engel, "Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017.

[49] L. Cominardi, L. M. Contreras, C. J. Bcrnardos, and I. Berberana, "Understanding qos applicability in 5g transport networks," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2018, pp. 1–5.

[50] D. Committee *et al.*, "Dsrc message communication minimum performance requirements: Basic safety message for vehicle safety applications," *SAE, Draft Std. J*, vol. 2945.

[51] G. Deng, X. Xie, L. Shi, and R. Li, "Hybrid information forwarding in vanets through named data networking," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 1940–1944.

[52] A. Detti, C. Pisa, and N. Blefari-Melazzi, "Multipath forwarding strategies in information centric networks with aimd congestion control," *arXiv preprint arXiv:1412.2204*, 2014.

[53] A. Detti, C. Pisa, and N. B. Melazzi, "Modeling multipath forwarding strategies in information centric networks," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2015, pp. 324–329.

[54] S. Din, A. Paul, and A. Rehman, "5g-enabled hierarchical architecture for software-defined intelligent transportation system," *Computer Networks*, vol. 150, pp. 81–89, 2019.

[55] F. Dressler, F. Klingler, C. Sommer, and R. Cohen, "Not all vanet broadcasts are the same: Context-aware class based broadcast," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 17–30, 2017.

[56] W. Drira and F. Filali, "A pub/sub extension to ndn for efficient data collection and dissemination in v2x networks," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. IEEE, 2014, pp. 1–7.

[57] X. Duan, Y. Liu, and X. Wang, "Sdn enabled 5g-vanet: Adaptive vehicle clustering and beamformed transmission for aggregated traffic," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 120–127, 2017.

[58] J. M. Duarte, E. Kalogeiton, R. Soua, G. Manzo, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, and G. A. Rizzo, "A multi-pronged approach to adaptive and context aware content dissemination in vanets," *Mobile Networks and Applications*, vol. 23, no. 5, pp. 1247–1259, 2018.

[59] D. Eckhoff, A. Brummer, and C. Sommer, "On the impact of antenna patterns on vanet simulation," in *2016 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2016, pp. 1–4.

[60] S. Eichler, "Performance evaluation of the ieee 802.11 p wave communication standard," in *2007 IEEE 66th Vehicular Technology Conference*. IEEE, 2007, pp. 2199–2203.

[61] A. El-Keyi, T. ElBatt, F. Bai, and C. Saraydar, "Mimo vanets: Research challenges and opportunities," in *2012 International conference on computing, networking and communications (ICNC)*. IEEE, 2012, pp. 670–676.

[62] M. Esler, J. Hightower, T. Anderson, and G. Borriello, "Next century challenges: data-centric networking for invisible computing: the portolano project at

the university of washington," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 256–262.

[63] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing information networking further: From psirp to pursuit," in *International Conference on Broadband Communications, Networks and Systems.* Springer, 2010, pp. 1–13.

[64] R. Frederick, S. L. Casner, V. Jacobson, and H. Schulzrinne, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, Jan. 1996. [Online]. Available: https://rfc-editor.org/rfc/rfc1889.txt

[65] B. Friedlander, "Using mimo to increase the range of wireless systems," in *Conference Record of the Thirty-Ninth Asilomar Conference onSignals, Systems and Computers, 2005.* IEEE, 2005, pp. 1404–1408.

[66] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

[67] M. S. Gast, *802.11 ac: a survival guide: Wi-Fi at gigabit and beyond.* " O'Reilly Media, Inc.", 2013.

[68] X. Ge, Z. Li, and S. Li, "5g software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, 2017.

[69] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *2014 IEEE world forum on internet of things (WF-IoT).* IEEE, 2014, pp. 241–246.

[70] S. Gräfling, P. Mähönen, and J. Riihijärvi, "Performance evaluation of ieee 1609 wave and ieee 802.11 p for vehicular communications," in *2010 Second International Conference on Ubiquitous and Future Networks (ICUFN).* IEEE, 2010, pp. 344–348.

[71] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Vanet via named data networking," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 410–415.

[72] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida, "Navigo: Interest forwarding by geolocations in vehicular named data networking," in *World*

*of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a.* IEEE, 2015, pp. 1–10.

[73] H. Hartenstein and K. Laberteaux, *VANET: vehicular applications and internetworking technologies.* Wiley Online Library, 2010, vol. 1.

[74] E. Hossain, G. Chow, V. C. Leung, R. D. McLeod, J. Mišić, V. W. Wong, and O. Yang, "Vehicular telematics over heterogeneous wireless networks: A survey," *Computer Communications*, vol. 33, no. 7, pp. 775–793, 2010.

[75] V. Inzillo, F. De Rango, A. F. Santamaria, and A. A. Quintana, "A new switched beam smart antenna model for supporting asymmetrical communications extending inet omnet++ framework," in *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS).* IEEE, 2017, pp. 1–7.

[76] R. Jmal and L. C. Fourati, "Content-centric networking management based on software defined networks: Survey," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1128–1142, 2017.

[77] M. Kakkasageri and S. Manvi, "Information management in vehicular ad hoc networks: A review," *journal of network and computer Applications*, vol. 39, pp. 334–350, 2014.

[78] E. Kalogeiton, T. Kolonko, and T. Braun, "A multihop and multipath routing protocol using ndn for vanets," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2017 16th Annual Mediterranean.* IEEE, 2017, pp. 1–8.

[79] E. Kalogeiton, Z. Zhao, and T. Braun, "Is sdn the solution for ndn-vanets?" in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2017 16th Annual Mediterranean.* IEEE, 2017, pp. 1–6.

[80] E. Kalogeiton and T. Braun, "Infrastructure-assisted communication for ndn-vanets," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM).* IEEE, 2018, pp. 1–10.

[81] ——, "Equipping ndn-vanets with directional antennas for efficient content retrieval," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC).* IEEE, 2020, pp. 1–8.

[82] ——, "On the impact of sdn for transmission power adaptation and fib population in ndn-vanets," in *2020 18th ACM International Symposium on Mobility Management and Wireless Access (MobiWac)*. ACM, 2020, pp. 1–10.

[83] E. Kalogeiton, D. Iapello, and T. Braun, "A geographical aware routing protocol using directional antennas for ndn-vanets," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 133–136.

[84] E. Kalogeiton, T. Kolonko, and T. Braun, "A topology-oblivious routing protocol for ndn-vanets," *Annals of telecommunications*, vol. 73, no. 9-10, pp. 577–587, 2018.

[85] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE communications surveys & tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

[86] O. Karrakchou, N. Samaan, and A. Karmouch, "A survey of routing mechanisms in icns," in *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 2018, pp. 1–6.

[87] A. A. Khan, M. Abolhasan, and W. Ni, "5g next generation vanets using sdn and fog computing framework," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–6.

[88] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, and A. Ksentini, "Named data networking in vehicular ad hoc networks: State-of-the-art and challenges," *IEEE Communications Surveys & Tutorials*, 2019.

[89] M. Khurana, C. Ramakrishna, and S. Panda, "Capacity enhancement using mu-mimo in vehicular ad hoc network," *International Journal of Applied Engineering Research*, vol. 12, no. 16, pp. 5872–5883, 2017.

[90] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless networks*, vol. 6, no. 4, pp. 307–321, 2000.

[91] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007, pp. 181–192.

[92] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[93] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[94] I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, "Towards software-defined vanet: Architecture and services," in *Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean*. IEEE, 2014, pp. 103–110.

[95] M. Kuai, X. Hong, and Q. Yu, "Density-aware delay-tolerant interest forwarding in vehicular named data networking," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2016, pp. 1–5.

[96] D. H. Lee, K. Thar, D. Kim, and C. S. Hong, "Efficient parallel multi-path interest forwarding for mobile user in ccn," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 390–394.

[97] P. Li, T. Zhang, C. Huang, X. Chen, and B. Fu, "Rsu-assisted geocast in vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 53–59, 2017.

[98] X. Liu, M. J. Nicolau, A. Costa, J. Macedo, and A. Santos, "A geographic opportunistic forwarding strategy for vehicular named data networking," in *Intelligent distributed computing IX*. Springer, 2016, pp. 509–521.

[99] C. V. Lopez, "Quality of service (qos) for information centric networks," Nov. 29 2016, uS Patent 9,509,631.

[100] X. Ma, J. Zhang, X. Yin, and K. S. Trivedi, "Design and analysis of a robust broadcast scheme for vanet safety-related services," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 46–61, 2011.

[101] A. Mahmood, B. Butler, and B. Jennings, "Towards efficient network resource management in sdn-based heterogeneous vehicular networks," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2018, pp. 813–814.

[102] G. Manzo, E. Kalogeiton, A. Di Maio, T. Braun, M. R. Palattella, I. Turcanu, R. Soua, and G. Rizzo, “Deepndn: Opportunistic data replication and caching in support of vehicular named data,” in *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM.* IEEE, 2020, pp. 1–10.

[103] C. D. Marsan, “The evolution of the internet,” https://www.networkworld.com/article/2870267/the-evolution-of-the-internet.html, 2009 (Accessed February 2020).

[104] ——, “Number of connected users on the internet,” https://www.internetworldstats.com/emarketing.htm, 2009 (Accessed February 2020).

[105] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 255–265.

[106] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang. (2016, November) ndnsim 2: An updated ndn simulator for ns-3. [Online]. Available: https://named-data.net/wp-content/uploads/2016/11/ndn-0028-2-ndnsim-v2.pdf

[107] G. Mauri, M. Gerla, F. Bruno, M. Cesana, and G. Verticale, “Optimal content prefetching in ndn vehicle-to-infrastructure scenario,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2513–2525, 2016.

[108] P. Mendes, R. C. Sofia, J. Soares, V. Tsaoussidis, S. Diamantopoulos, and C.-A. Sarros, “Information-centric routing for opportunistic wireless networks,” in *Proceedings of the 5th ACM Conference on Information-Centric Networking.* ACM, 2018, pp. 194–195.

[109] R. I. Meneguette and L. H. Nakamura, “A flow control policy based on the class of applications of the vehicular networks,” in *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, 2017, pp. 137–144.

[110] Z. H. Mir and F. Filali, “Lte and ieee 802.11 p for vehicular networking: a performance evaluation,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 89, 2014.

[111] S. Moser, L. Behrendt, and F. Slomka, "Mimo-enabling phy layer enhancement for vehicular ad-hoc networks," in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2015, pp. 142–147.

[112] G. Naik, B. Choudhury, and J.-M. Park, "Ieee 802.11 bd & 5g nr v2x: Evolution of radio access technologies for v2x communications," *IEEE Access*, vol. 7, pp. 70 169–70 184, 2019.

[113] D. Nguyen, M. Fukushima, K. Sugiyama, and A. Tagami, "Efficient multipath forwarding and congestion control without route-labeling in ccn," in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 1533–1538.

[114] L. Nkenyereye, C. H. Liu, and J. Song, "Towards secure and privacy preserving collision avoidance system in 5g fog based internet of vehicles," *Future Generation Computer Systems*, vol. 95, pp. 488–499, 2019.

[115] E. Perahia and R. Stacey, *Next generation wireless LANs: 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[116] D. Perino and M. Varvello, "A reality check for content centric networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. ACM, 2011, pp. 44–49.

[117] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Internet Requests for Comments, RFC 3561, July 2003.

[118] N. Raza, S. Jabbar, J. Han, and K. Han, "Social vehicle-to-everything (v2x) communication model for intelligent transportation systems based on 5g scenario," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*. ACM, 2018, p. 54.

[119] G. Remy, S.-M. Senouci, F. Jan, and Y. Gourhant, "Lte4v2x: Lte for a centralized vanet organization," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011, pp. 1–6.

[120] B. Ribeiro, F. Gonçalves, A. Santos, M. J. Nicolau, B. Dias, J. Macedo, and A. Costa, "Simulation and testing of a platooning management protocol implementation," in *International Conference on Wired/Wireless Internet Communication*. Springer, 2017, pp. 174–185.

[121] G. F. Riley and T. R. Henderson, “The ns-3 network simulator,” *Modeling and tools for network simulation*, pp. 15–34, 2010.

[122] G. Rossini and D. Rossi, “Evaluating ccn multi-path interest forwarding strategies,” *Computer Communications*, vol. 36, no. 7, pp. 771–778, 2013.

[123] L. Rui, H. Guo, H. Huang, R. Shi, and X. Qiu, “Mupf: Multiple unicast path forwarding in content-centric vanets,” *Ad Hoc Networks*, vol. 81, pp. 211–225, 2018.

[124] C.-A. Sarros, V. Demiroglou, and V. Tsaoussidis, “Intermittently-connected iot devices: Experiments with an ndn-dtn architecture,” in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–9.

[125] C.-A. Sarros, S. Diamantopoulos, S. Rene, I. Psaras, A. Lertsinsrubtavee, C. Molina-Jimenez, P. Mendes, R. Sofia, A. Sathiaseelan, G. Pavlou *et al.*, “Connecting the edges: A universal, mobile-centric, and opportunistic communications architecture,” *IEEE Communications Magazine*, vol. 56, no. 2, pp. 136–143, 2018.

[126] J. Schaerer, Z. Zhao, J. Carrera, S. Zumbrunn, and T. Braun, “Sdnwisebed: A software-defined wsn testbed,” in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2019, pp. 317–329.

[127] E. Schiller, E. Kalogeiton, T. Braun, A. Gomes, and N. Nikaein, “Icn/dtn for public safety in mobile networks,” in *Wireless Public Safety Networks 3*. Elsevier, 2017, pp. 231–247.

[128] E. Schiller, N. Nikaein, E. Kalogeiton, M. Gasparyan, and T. Braun, “Cds-mec: Nfv/sdn-based application management for mec in 5g systems,” *Computer Networks*, vol. 135, pp. 96–107, 2018.

[129] T. Shinkawa, T. Terauchi, T. Kitani, N. Shibata, K. Yasumoto, M. Ito, and T. Higashino, “A technique for information sharing using inter-vehicle communication with message ferrying,” in *7th International Conference on Mobile Data Management (MDM’06)*. IEEE, 2006, pp. 130–130.

[130] F. Siddiqui, S. Zeadally, and K. Salah, “Gigabit wireless networking with ieee 802.11 ac: technical overview and challenges,” *Journal of Networks*, vol. 10, no. 3, p. 164, 2015.

[131] C. Sommer and F. Dressler, "Using the right two-ray model? a measurement based evaluation of phy models in vanets," in *Proc. ACM MobiCom*, 2011, pp. 1–3.

[132] C. Sommer, D. Eckhoff, and F. Dressler, "Ivc in cities: Signal attenuation by buildings and how parked cars can improve the situation," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1733–1745, 2013.

[133] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of ieee 802.11 p radio shadowing in urban environments," in *2011 Eighth international conference on wireless on-demand network systems and services*. IEEE, 2011, pp. 84–90.

[134] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of two-ray path loss models for vehicular network simulation," in *2012 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2012, pp. 64–69.

[135] J. Son, D. Kim, H. S. Kang, and C. S. Hong, "Forwarding strategy on sdn-based content centric network for efficient content delivery," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 220–225.

[136] R. Soua, E. Kalogeiton, G. Manzo, J. M. Duarte, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, and G. A. Rizzo, "Sdn coordination for ccn and fc content dissemination in vanets," in *Ad Hoc Networks*. Springer, 2017, pp. 221–233.

[137] Q. Sun, W. Wendong, Y. Hu, X. Que, and G. Xiangyang, "Sdn-based autonomic ccn traffic management," in *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2014, pp. 183–187.

[138] S. Tiennoy and C. Saivichit, "Using a distributed roadside unit for the data dissemination protocol in vanet with the named data architecture," *IEEE Access*, vol. 6, pp. 32 612–32 623, 2018.

[139] C. K. Toh, *Ad hoc mobile wireless networks: protocols and systems*. Pearson Education, 2001.

[140] M. Tortelli, L. A. Grieco, G. Boggia, and K. Pentikousisy, "Cobra: Lean intra-domain routing in ndn," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 839–844.

[141] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 26–33, 2010.

[142] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1202–1207.

[143] V. Tsaoussidis and I. Matta, "Open issues on tcp for mobile computing," *Wireless Communications and Mobile Computing*, vol. 2, no. 1, pp. 3–20, 2002.

[144] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An on-demand multi-path interest forwarding strategy for content retrievals in ccn," in *2014 IEEE network operations and management symposium (NOMS)*. IEEE, 2014, pp. 1–6.

[145] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi, and A. Detti, "Supporting information-centric functionality in software defined networks," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 6645–6650.

[146] A. Wahid, M. A. Shah, F. F. Qureshi, H. Maryam, R. Iqbal, and V. Chang, "Big data analytics for mitigating broadcast storm in vehicular content centric networks," *Future Generation Computer Systems*, 2017.

[147] T. Wang, B. Ai, R. He, and Z. Zhong, "Two-dimension direction-of-arrival estimation for massive mimo systems," *IEEE Access*, vol. 3, pp. 2122–2128, 2015.

[148] X. Wang and X. Wang, "Vehicular content-centric networking framework," *IEEE Systems Journal*, 2018.

[149] T.-J. Wu, W. Liao, and C.-J. Chang, "A cost-effective strategy for road-side unit placement in vehicular networks," *IEEE Transactions on Communications*, vol. 60, no. 8, pp. 2295–2303, 2012.

[150] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2014.

[151] X. Xie, B. Huang, S. Yang, and T. Lv, "Adaptive multi-channel mac protocol for dense vanet with directional antennas," in *2009 6th IEEE Consumer Communications and Networking Conference*. IEEE, 2009, pp. 1–5.

[152] X. Xie, F. Wang, K. Li, P. Zhang, and H. Wang, "Improvement of multi-channel mac protocol for dense vanet with directional antennas," in *2009 IEEE Wireless Communications and Networking Conference*. IEEE, 2009, pp. 1–6.

[153] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2013.

[154] L. Yang, W. Xiang, and G. Zheng, "A study on vehicle connectivity for vanets adopting beamforming antenna array," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[155] I. Yaqoob, I. Ahmad, E. Ahmed, A. Gani, M. Imran, and N. Guizani, "Overcoming the key challenges to establishing vehicular communication: Is sdn the answer?" *IEEE Communications Magazine*, vol. 55, no. 7, pp. 128–134, 2017.

[156] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM computer communication review*, vol. 42, no. 3, pp. 62–67, 2012.

[157] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): challenges and perspectives," in *2006 6th International Conference on ITS Telecommunications*. IEEE, 2006, pp. 761–766.

[158] Y.-T. Yu, Y. Li, X. Ma, W. Shang, M. Sanadidi, and M. Gerla, "Scalable opportunistic vanet content routing with encounter information," in *2013 21st IEEE international conference on network protocols (ICNP)*. IEEE, 2013, pp. 1–6.

[159] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[160] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu, "A transport protocol for content-centric networking with explicit congestion control," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2014, pp. 1–8.

[161] G. Zhang, Y. Xu, X. Wang, X. Tian, J. Liu, X. Gan, H. Yu, and L. Qian, "Multicast capacity for vanets with directional antenna and delay constraint," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 818–833, 2012.

[162] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[163] Y. Zhang, J. Liu, T. Huang, J. Chen, and Y. Liu, "Multi-path interests routing scheme for multi-path data transfer in content centric networking," 2013.

[164] Z. Zhao, M. Karimzadeh, F. Gerber, and T. Braun, "Mobile crowd location prediction with hybrid features using ensemble learning," *Future Generation Computer Systems*, 2018.

[165] Z. Zhao, E. Schiller, E. Kalogeiton, T. Braun, B. Stiller, M. T. Garip, J. Joy, M. Gerla, N. Akhtar, and I. Matta, "Autonomic communications in software-driven networks," *IEEE journal on selected areas in communications*, vol. 35, no. 11, pp. 2431–2445, 2017.

[166] J. Zheng, Y. Wang, and N. L. Nihan, "Tracking vehicles with gps: Is it a feasible solution?" 2002.

[167] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: architecture and challenges," *IEEE Network*, vol. 30, no. 4, pp. 72–80, July 2016.

[168] M. Zhu, J. Cao, D. Pang, Z. He, and M. Xu, "Sdn-based routing for efficient message propagation in vanet," in *International Conference on Wireless Algorithms, Systems, and Applications.* Springer, 2015, pp. 788–797.

[169] H. Zimmermann, "The cyclades experience-results and impacts." in *IFIP Congress*, 1977, pp. 465–469. [Online]. Available: http://dblp.uni-trier.de/db/conf/ifip/ifip1977.html#Zimmermann77

# Declaration of consent

on the basis of Article 30 of the RSL Phil.-nat. 18

Name/First Name:

Registration Number:

Study program:

Bachelor ☐ Master ☐ Dissertation ☐

Title of the thesis:

Supervisor:

I declare herewith that this thesis is my own work and that I have not used any sources other than those stated. I have indicated the adoption of quotations as well as thoughts taken from other authors as such in the thesis. I am aware that the Senate pursuant to Article 36 paragraph 1 litera r of the University Act of 5 September, 1996 is authorized to revoke the title awarded on the basis of this thesis.

For the purposes of evaluation and verification of compliance with the declaration of originality and the regulations governing plagiarism, I hereby grant the University of Bern the right to process my personal data and to perform the acts of use this requires, in particular, to reproduce the written thesis and to store it permanently in a database, and to use said database, or to make said database available, to enable comparison with future theses submitted by others.

Place/Date

Signature

# Eirini Kalogeiton | CV

eirini.kalogeiton@inf.unibe.ch

## Education

### Academic Qualifications

- **PhD in Computer Science** — **Bern, Switzerland**
  *Communication and Distributed Systems Group, Institute of Computer Science, University of Bern, Switzerland* — *2016-2020*
- **Master of Science** — **Xanthi, Greece**
  *Department of Electrical and Computer Engineer, Democritus University of Thrace, Xanthi, Greece* — *2014-2016*
- **Diploma in Electrical and Computer Engineer (5-year programme)** — **Xanthi,Greece**
  *Department of Electrical and Computer Engineer, Democritus University of Thrace, Xanthi, Greece* — *2008-2014*

## Publications during the PhD:

- **E. Kalogeiton**, T. Braun: "On the Impact of SDN for Transmission Power Adaptation and FIB Population in NDN-VANETs", 18th ACM International Symposium on Mobility Management and Wireless Access (MobiWac), ACM, 2020.
- G. Manzo, **E. Kalogeiton**, A. Di Maio, T. Braun, M. R. Palattella, I. Turcanu, R. Soua, G. Rizzo: "DeepNDN: Opportunistic Data Replication and Caching in Support of Vehicular Named Data", IEEE 21st International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2020.
- **E. Kalogeiton**, D. Iapello, T. Braun: "Equipping NDN-VANETs with Directional Antennas for Efficient Content Retrieval", IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2020.
- **E. Kalogeiton**, D. Iapello, T. Braun: "A Geographical Aware Routing Protocol using Directional Antennas for NDN-VANETs", IEEE 44th Conference on Local Computer Networks (LCN), IEEE, 2019.
- J.M. Duarte, **E. Kalogeiton**, R. Soua, G. Manzo, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, G. Rizzo: "A multi-pronged approach to adaptive and context aware content dissemination in VANETs", Mobile Networks and Applications, vol 23, no 5, pp 1247-1259, Springer, 2018.
- **E. Kalogeiton**, T. Kolonko, T. Braun: "A topology-oblivious routing protocol for NDN-VANETs", Annals of Telecommunications, vol 73, no 9-10, pp 577-587, Springer, 2018.
- E. Schiller, N. Nikaein, **E. Kalogeiton**, M. Gasparyan, T.Braun: "CDS-MEC: NFV/SDN-based application management for MEC in 5G systems", Computer Networks, vol 135, pp 96-107, Elsevier, 2018
- **E. Kalogeiton**, T. Braun: "Infrastructure-assisted communication for NDN-VANETs", IEEE 19th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2018.
- Z. Zhao, E. Schiller, **E. Kalogeiton**, T. Braun, B. Stiller, M. T. Garip, J. Joy, M. Gerla, N. Akhtar, I. Matta: "Autonomic communications in software-driven networks", IEEE Journal on Selected Areas in Communications (JSAC), vol 35, no 11, pp 2431-2445, IEEE, 2017.
- **E. Kalogeiton**, T. Kolonko, T. Braun: "A multihop and multipath routing protocol using NDN for VANETs", 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net). IEEE, 2017.
- **E. Kalogeiton**, Z. Zhao, T. Braun: "Is SDN the solution for NDN-VANETs?", 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net). IEEE, 2017.
- E. Schiller, **E. Kalogeiton**, T. Braun, A. Gomes, N. Nikaein: "ICN/DTN for public safety in mobile networks", In Wireless Public Safety Networks, vol 3, pp 231-247, Elsevier, 2017.
- R. Soua, **E. Kalogeiton**, G. Manzo, J. M. Duarte, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, G. Rizzo: "SDN coordination for CCN and FC content dissemination in VANETs", Ad Hoc Networks, pp 221-233, Springer, 2017.